UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

NUEVO SISTEMA EMPÍRICO DE APOYO A LA TOMA DE DECISIONES DE COMPRAVENTA DE ACCIONES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELECTRICISTA

LUIS IGNACIO MORENO ARACENA

PROFESOR GUÍA:
RICHARD WEBER HAAS

MIEMBOS DE LA COMISIÓN:
CRISTIÁN BRAVO ROMÁN
MARCOS ORCHARD CONCHA

SANTIAGO DE CHILE
2014

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRICISTA
POR: LUIS IGNACIO MORENO ARACENA
FECHA: ABRIL 2014
PROF. GUÍA: Sr. RICHARD WEBER

"NUEVO SISTEMA EMPÍRICO DE APOYO A LA TOMA DE DECISIONES DE COMPRAVENTA DE ACCIONES"

En el mundo financiero, la decisión de compraventa de activos se suele asentar en el análisis fundamental a largo plazo, combinado con análisis técnico a corto plazo; con el objetivo de establecer un momento adecuado para la adquisición y enajenación de activos.

En la última década, se ha verificado un crecimiento exponencial en la capacidad de procesamiento y de manejo de bases de datos; siendo la minería de estos vastamente estudiada y aplicada exitosamente en distintos campos, entre los cuales se encuentran las finanzas. En el presente trabajo, se estudia la existencia de estructura con capacidad predictiva en activos financieros, con el fin de anticipar cambios de tendencia y así obtener retornos por sobre el mercado. Para esto, se desarrolla a cabalidad el proceso de extracción de conocimiento de bases de datos, el que considera desde la generación de variables, hasta la obtención de información, a partir de los datos transaccionales de las acciones que componen el Índice de Precios Selectivo de Acciones (IPSA) 2013.

En este sentido, es importante precisar que la metodología clásica en la predicción de series de tiempo, se basa en la utilización de precios anteriores para así predecir el precio futuro, utilizando ventanas de tiempo estáticas. En este trabajo se estudia un método nuevo, donde la variable objetivo, en vez de ser retornos en ventanas temporales, son tanto retornos como ventanas dinámicas, extraídas a partir de extensiones no causales de retracciones porcentuales del precio (indicador ZigZag) de las acciones, las que representan mínimos y máximos locales de la serie de tiempo; evitando así sobreajuste temporal y acomodándose a los cambios de ciclo del activo en estudio.

Se generan variables independientes a partir de datos de transacciones realizadas por parte de miembros de las compañías (Insiders) e indicadores técnicos tales como cruces, divergencias y zonas de agotamiento a partir de Medias Móviles Convergentes/Divergentes, Índice de fuerza Relativa y Oscilador Estocástico. Se realiza selección de características mediante *Forward Selection* y *Backward Elimination*, para encontrar un subconjunto de atributos adecuado y analizar su impacto predictivo. Se aplican algoritmos de aprendizaje supervisado con capacidad de extraer patrones altamente no lineales, destacando Redes Neuronales de Retropropagación, Máquinas de Soporte Vectorial y Métodos Basados en Similitud. Con el fin de determinar el ciclo del mercado al que mejor se ajustan los atributos extraídos y el mejor modelo predictor sobre la base de datos no balanceada, se evalúa la combinación de predicciones de compraventa (anticipaciones de cambio de tendencia) utilizando clasificador Bayesiano ingenuo y operadores lógicos.

Finalmente, se realiza una evaluación tanto cualitativa (visual) como cuantitativa (mediante un simulador de inversiones) del comportamiento de las recomendaciones de compraventa; analizando la distribución de retorno, *drawdown* y tiempo de apertura de las operaciones. De lo anterior puede concluirse que dentro de lo caótico del mercado bursátil, subyace estructura altamente no lineal con poder anticipativo de cambios de tendencia de los activos; la cual se puede atribuir a que, en Chile, el mercado es poco profundo, ilíquido o ineficiente.

… “*Dedicado a mi familia, quienes siempre me han apoyado incondicionalmente”*…

Tabla de contenido

Índice de tablas

Índice de ilustraciones

Capítulo 1

Introducción

Este estudio está orientado a determinar si existen factores que anticipen cambios de tendencia en las acciones que componen el Índice de Precios Selectivo de Acciones (IPSA) 2013 (Anexo A) en la Bolsa de Comercio de Santiago, Chile. Para esto, el trabajo realizado contempla a cabalidad el proceso de extracción de conocimiento en bases de datos [1] conocido como *Knowledge Discovery in Databases* (KDD); en conjunto con la implementación de indicadores técnicos y algoritmos involucrados. Se finaliza realizando una evaluación realista, mediante un simulador de inversiones, de potenciales ganancias y riesgos al operar las predicciones de compraventa o anticipaciones de cambio de tendencia.

La investigación realizada consiste en el análisis inicial de una metodología con valor agregado en la predicción del mercado bursátil, la cual puede ser mejorada en diferentes aspectos y evaluada en otros tipos de mercados.

1.1. Motivación

Desde el nacimiento del mercado financiero, grupos privados y gubernamentales intentan obtener utilidades de éste. Para esto, se han desarrollado teorías y herramientas que permiten analizar los instrumentos financieros, con el objetivo de obtener ventaja competitiva en la transacción de sus activos.

Alrededor del mundo, los países crean sus propias bolsas de comercio. Bancos, universidades y sociedades particulares los analizan y estudian, encontrando muchas veces que el mercado es aleatorio. Sin embargo, esta memoria se basa en la hipótesis de que el mercado, bajo ciertas circunstancias, es caótico, es decir, con movimientos aparentemente aleatorios, pero con estructuras o patrones de comportamiento. Por esto, este estudio postula que no se considera que las acciones evaluadas posean comportamiento de “paseo aleatorio” [2].

Con la entrada de Internet, los inversionistas ya no necesitan tranzar desde el mismo lugar físico. En conjunto con la llegada de computadores más potentes, comienza el estudio de la extracción de conocimiento a partir de bases de datos, dando cabida a la predicción de instrumentos financieros mediante herramientas computacionales.

Existen múltiples intentos de identificar patrones de comportamiento para posteriormente estimar y proyectar sus efectos en el mercado, tales como: Identificar cuándo una firma va a quebrar, como lo hacen Odon y Sharda [3] o Min y Lee [4], o en general, cómo información liberada al mercado altera los precios de activos de mercado de forma sistemática, como lo muestran Del Brio et al. [5] en el mercado español, o de forma general Bhattacharya y Daouk [6]. Además se han realizado trabajos en *commodities* [7] y derivados financieros, como los trabajos de Schwartz [8], Yang, Bessler y Leathman [9]; o el de Donoho [10] para el caso de opciones de mercado.

Por lo mismo, es necesario utilizar técnicas o herramientas que sean capaces de encontrar variables adicionales a las transaccionales para determinar patrones, utilizando métodos de minería de datos.

1.2. Alcances

El estudio se desarrolla con el objetivo de anticipar cambios de tendencia de activos, con el fin de extraer rentabilidades por sobre el mercado. Para esto, se desea estudiar la existencia de estructura a partir de atributos extraídos de la serie de tiempo financiera y datos transaccionales de personas que trabajan para la empresa (Insiders).

El proceso requiere implementar distintos códigos. Para esto, se utiliza particularmente MATLAB R2013a, aplicando indicadores a las acciones en estudio y extrayendo atributos para cada período de tiempo. Se aplican los modelos de aprendizaje supervisado y se evalúan las recomendaciones de compraventa (anticipaciones de cambio de tendencia) mediante un simulador de inversiones, siendo necesario implementar:

- Indicadores técnicos:
  - Medias Móviles.
  - Medias Móviles Convergentes/Divergentes.
  - Índice de fuerza Relativa.
  - Oscilador Estocástico.
  - ZigZag.
- Extracción de variables:
  - Cruces.
  - Agotamiento.
  - Divergencia.
  - Insider.
  - Extensión de ZigZag.

- Modelos de aprendizaje supervisado:
  - Regresión Lineal.
  - Red Neuronal de Retropropagación (BPNN).
  - Máquinas de Soporte Vectorial (SVM).
  - Métodos multivariados basados en Similitud (SBM).
  - Clasificador Bayesiano ingenuo (NB).
- Simulador de Inversiones.

1.3. Objetivos

1.3.1. Objetivo general

El objetivo general de la memoria consiste en identificar si subyace estructura en el mercado financiero chileno, que permita anticipar cambios de tendencia de un activo.

1.3.2. Objetivos específicos

- Implementar indicadores que describan cuantitativamente el estado de un activo, que reflejen características de: fuerza, dirección y momento.
- Extracción de variables a partir de indicadores técnicos y precio de activo, que permitan aplicar minería de datos.
- Generación de atributos a partir de datos transaccionales por parte de Insiders.
- Creación de variable objetivo que refleje mínimos y máximos locales de la serie temporal de precio de un activo, con capacidad de etiquetado y posterior aplicación de herramientas de minería de datos.
- Selección de subconjunto de atributos mediante *Forward Selection* y *Backward Propagation*.
- Facilitar la visualización de la selección de atributos, con el fin de identificar su poder predictivo.
- Implementación de herramientas de minería de datos (aprendizaje supervisado) que permitan reflejar los atributos extraídos a las variables objetivo.
- Implementar simulador realista que permita evaluar cuantitativamente el poder anticipativo de la metodología propuesta.

1.4. Estructura de la memoria

La estructura utilizada en esta memoria es la siguiente:

- **Capítulo 1. Introducción**: Se introduce y motiva al lector en el tema investigado y se presentan los alcances, objetivos y estructura de la memoria.

- **Capítulo 2. Contextualización:** Se explican los conceptos necesarios para la contextualización y comprensión del trabajo, abarcando temas tanto financieros como de extracción de conocimiento a partir de bases de datos.

- **Capítulo 3. Generación de base de datos:** Se presentan los detalles de la generación de atributos de datos transaccionales de Insiders, extracción de variables independientes a partir de indicadores técnicos y generación de la variable dependiente. Finalmente, se prepara la base de datos a ser utilizada en la fase de minería de datos.

- **Capítulo 4. Aplicación de minería de datos:** Se realiza selección de características y se aplican modelos de aprendizaje supervisado a las bases de datos previamente preparadas. Se analiza la capacidad predictiva de las variables extraídas, se establece el ciclo de mercado al que mejor se adaptan los atributos y se selecciona el mejor modelo de aprendizaje supervisado (o combinación de estos).

- **Capítulo 5. Análisis de predicciones de compraventa:** Se lleva a cabo un análisis cualitativo y cuantitativo de los cambios de tendencia predichos. Para esto, se visualizan las predicciones de compraventa y se analiza la ganancia (a partir de retorno y duración de operaciones) y riesgo (a partir de *drawdown*), mediante un simulador de inversiones.

- **Capítulo 6. Conclusiones:** Se enumeran las conclusiones de la memoria realizada y se propone trabajo futuro.

Capítulo 2

Contextualización

El objetivo del presente capítulo es ubicar al lector en el entorno en el cual se desarrolla este trabajo de título.

2.1. Mercado financiero

El mercado financiero es un espacio que permite la transacción o intercambio de instrumentos financieros, en el cual se definen sus precios por las fuerzas de oferta y demanda. Éste se ve afectado por distintos factores, ya que la oferta y demanda van directamente relacionadas por el comportamiento de los inversionistas, los que tranzan estos instrumentos basados en diversos análisis, con el fin de extraer utilidades de sus operaciones.

Uno de los instrumentos más conocidos son las acciones, que corresponden a títulos emitidos por una sociedad y representan el valor de una de las fracciones iguales en que se divide su capital social y se abrevian por un nemotécnico bursátil, el cual se suele llamar "código" o *ticker*.

Las acciones generalmente se componen en índices, cuyo objetivo es clasificar un conjunto de activos representativo de un rubro o de un sector bursátil; otra clasificación común es por capitalización bursátil, definiéndose ésta en (2.1) para una acción en el momento $\boldsymbol{t}$, con $t > 0$:

$$CapitalizaciónBursátil_t = CantidadTotalDeAcciones_t \cdot PrecioAcción_t \quad \textbf{(2.1)}$$

En Chile, el índice más utilizado es el Índice de Precios Selectivos de Acciones o IPSA, este considera las 40 acciones chilenas con mayor presencia bursátil de la Bolsa de Comercio de Santiago, su composición para el año 2013 se muestra en Anexo A.

En el mundo financiero se introducen términos respecto al rendimiento de compraventa de acciones, siendo los más importantes en el contexto de esta memoria:

- Retorno
- *Drawdown*
- *Stop-loss*
- *Take-profit*.

En finanzas, Retorno o *Return* es la ganancia sobre una inversión, éste corresponde a la variación porcentual de un activo financiero dentro de los períodos de interés. En este sentido, una pérdida en vez de ganancia, se describe como un retorno negativo. En (2.2) se expresa el retorno en un período $\boldsymbol{t}$ respecto a $\boldsymbol{k}$ períodos anteriores.

$$Retorno(t,k) = \frac{Precio_t - Precio_{t-k}}{Precio_{t-k}} \qquad \textbf{(2.2)}$$

El *drawdown* o retroceso de la curva de resultados, representa la pérdida respecto al máximo histórico en la curva de capital o *equity* (resultado de transacciones bursátiles). Éste se utiliza para medir el riesgo de un sistema de inversiones, se puede medir en valor neto o porcentual, y se considera hasta que el capital supere el último máximo.

Johansen y Sornette [11] discuten que el *drawdown* representa una medida más natural del riesgo real del mercado, cualidad que no poseen la varianza u otras medidas de distribución centrada de retornos. Por otra parte, Chekhlov, Uryasev y Zabarankin [12] consideran que ésta corresponde a una generalización de medida de desviación a un caso dinámico. En la Ilustración 2.1 se ejemplifica un *drawdown* de 15% en la curva de ganancias.

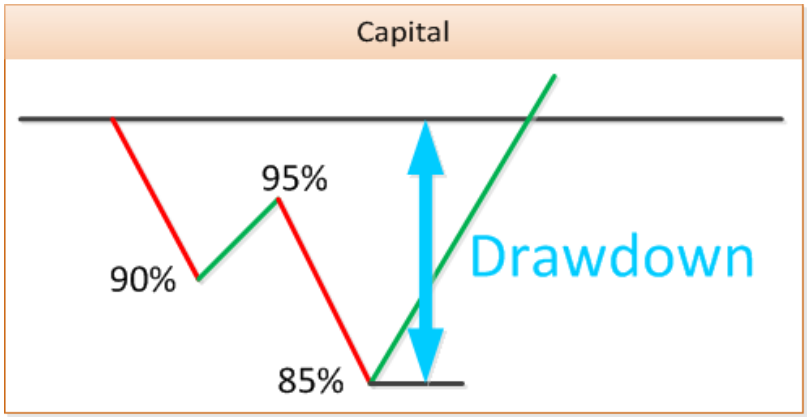

Ilustración 2.1: Ejemplo *drawdown* de 15%

Stop-loss (SL) y *take-profit* (TP) corresponden a precios determinados en que se cierra una operación bursátil cuando el precio va en contra y a favor respectivamente. Estos valores pueden ser estáticos o dinámicos (varían en el transcurso de una operación), siendo los estáticos más utilizados debido a su

sencillez. Acar y Toffel [13] discuten resultados empíricos de su utilización, destacando que mientras estrategias de *take-profit* pueden ser un lujo, estrategias de *stop-loss* son completamente necesarias, poniendo énfasis en proteger ganancias y limitar pérdidas a cero.

2.2. Especulación

La especulación (en economía) corresponde a la predicción del movimiento futuro del precio de un instrumento financiero, con el fin de extraer ganancias de las transacciones financieras. Nicholas Kaldor en Castañeda y Hernández [14] la define como “la compra (o venta) de bienes con vistas a su posterior reventa (recompra), cuando el motivo de tal acción es la expectativa de un cambio en los precios afectados con respecto al precio dominante y no la ganancia derivada de su uso, o de algún tipo de transformación efectuada sobre éstos o de la transferencia entre mercados distintos”. Además, se discute la incorporación a la definición, la intencionalidad de los agentes por obtener valor agregado, y en particular en cuanto a obtener elevada rentabilidad y rapidez en la liquidación de la ganancia.

En general existen dos tipos de análisis predictivos o especulativos, estos son análisis fundamental y análisis técnico.

2.2.1. Análisis fundamental

Análisis fundamental consiste en la estimación del valor fundamental o esencial de un título o acción, para luego comparar éste con el precio actual del activo. La hipótesis fundamental asume que si el precio actual o de mercado de un activo es superior al valor fundamental, la acción está sobrevalorada y su precio disminuirá en el futuro a través de ajustes del mercado.

En el análisis fundamental se tiende a utilizar tanto propiedades intrínsecas o fundamentales, como información extrínseca en la evaluación del valor fundamental. Dentro de las propiedades intrínsecas se debe considerar cualquier tipo de información que afecte el valor de un título, dentro de las cuales se puede ponderar: Estados financieros, previsiones económicas, técnicas de evaluación de empresas y cualquier información económica en general. La información extrínseca generalmente corresponde a noticias, en particular sobre acontecimientos de origen político, social y económico, poniendo en contraste cómo este tipo de información ha afectado históricamente la cotización de la acción, pudiendo así estimar su valor esencial y anticipar un posible movimiento futuro.

Otro tipo de información que puede otorgar valor agregado es utilizar los datos transaccionales de personas que trabajan en una sociedad con acciones, o que poseen una gran cantidad de activos de una sociedad; éstos pueden poseer información privilegiada sobre la compañía, y así concretar operaciones de adquisición o enajenación antes que la información salga a nivel público; este tipo de persona se llama Insider.

Debido a que la información de compra y venta por parte de Insiders es de conocimiento público (regulado por la Superintendencia de Valores y Seguros de Chile <SVS>), ésta puede ser utilizada para buscar una relación entre la compraventa de acciones por parte de Insiders con respecto a un cambio de tendencia de la respectiva acción a futuro, haciendo posible integrar la información de éstos en una estrategia de inversión con el objetivo de mejorar la rentabilidad del sistema.

2.2.2. Análisis técnico

Análisis técnico es parte de las herramientas utilizadas en la toma de decisiones por parte de agentes participantes del mercado bursátil. De acuerdo a Neely [15] en el mercado de divisas, y en general como expone Neely [16], sus tres premisas son:

i. La acción del mercado lo descuenta todo. Esta premisa es el cimiento de análisis técnico, significa que todo lo que pueda afectar el precio de un activo, ya sea de índole político, económico o psicológico, está reflejado en el precio.
ii. Los precios se mueven por tendencias. El propósito del análisis técnico es identificar una nueva tendencia en su fase de desarrollo, con el fin de operar a favor de esta. Por corolario, es más probable que la tendencia continúe en vez de revertirse.
iii. La historia tiende a repetirse. Basado en la psicología humana, esta premisa supone que si se han identificado patrones de tendencia durante décadas en el pasado. Si han funcionado bien en el pasado, continuarán funcionando en el futuro.

Dentro del análisis técnico se encuentra análisis de volumen y análisis cuantitativo, el primero de estos consta de utilizar la información de volumen de transacciones para cada período y así identificar cuándo se realizan grandes compras o ventas de acciones con el fin de identificar una nueva tendencia. Muchas veces este tipo de análisis se considera aparte de análisis técnico, pues considera información que no se desprende directamente del precio.

Por otra parte existe el análisis cuantitativo, éste se puede considerar parte o un derivado directo de análisis técnico y corresponde a la utilización de

matemática financiera para llevar a cabo análisis predictivos. En esta área se tienden a utilizar herramientas derivadas de la física y estadística sobre la serie temporal financiera. Estos mecanismos serían capaces de anticipar movimientos de los activos en el corto plazo.

Dentro de la serie temporal se suelen considerar distintos tipos de precios para cada período, los más comunes son *Open*, *High*, *Low* y *Close* (OHLC) para el precio de apertura, máximo, mínimo y de cierre respectivamente. Estos precios se suelen graficar para cada período como "línea del oeste" o "vela japonesa", siendo este último el más utilizado por su fácil de comprensión visual (ver Ilustración 2.2).

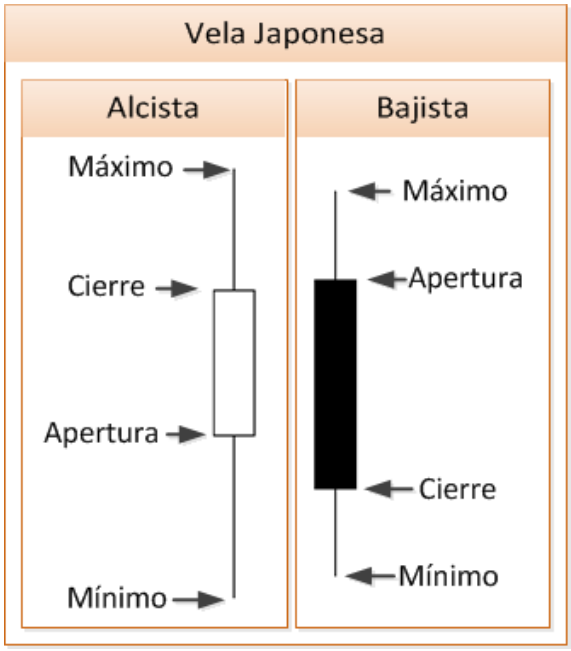

Ilustración 2.2: Vela japonesa como ejemplo de OHLC

2.2.3. Indicadores técnicos

Los indicadores técnicos corresponden a una serie temporal numérica extraída del precio de una acción. Entre los indicadores más utilizados se encuentran las categorías de tendencia y osciladores.

Los indicadores de tendencia, como su nombre lo indica, intentan extraer la tendencia del activo a través de su precio, filtrando así retornos cambiarios a corto plazo que no sean de interés. Dentro de las herramientas de análisis técnico, la media móvil (*Moving Average* <MA>) es una de las herramientas de tendencia más utilizada [17] [18].

Los osciladores intentan anticipar cambios en la tendencia del activo a corto plazo. Éstos se basan en normalizaciones del precio del activo con el fin de obtener la fase oscilatoria actual. Dentro de los osciladores más utilizados se encuentran: Medias Móviles Convergentes/Divergentes (*Moving Average Convergence/Divergence* <MACD>), Índice de Fuerza Relativa (*Relative Strength Index* <RSI>) y Oscilador Estocástico (*Stochastic Oscillator* <SO>).

2.2.3.1. Medias móviles

Las medias móviles son la categoría más utilizada de indicadores de tendencia, éstas corresponden a una serie temporal, donde cada valor representa al promedio ponderado de un subconjunto de datos. Dentro de las medias móviles existen diferentes categorías, sin embargo, las más utilizadas debido a su simplicidad, son las medias móviles simples y exponenciales. En la Ilustración 2.3 se ejemplifican ambos tipos de MA con suavizado de 34 y 200 períodos aplicado al precio de cierre.

Ilustración 2.3: Ejemplos de medias móviles

2.2.3.1.1. Media móvil simple

La media móvil simple (*Simple Moving Average* <SMA>) pertenece a la categoría de filtros con respuesta finita el impulso [19] (Finite Impulse Response <FIR>), es decir, utiliza un número finito de términos no nulos en su cálculo. En aplicaciones financieras, corresponde al promedio no ponderado de los últimos $\boldsymbol{n}$ períodos del tipo de precio de interés.

Si se representa la serie temporal como $x_t, x_{t-1}, \dots, x_{t-(n-1)}$, siendo $\boldsymbol{t}$ el momento a calcular, y $\boldsymbol{n}$ los períodos de suavizado a considerar, luego la SMA se puede calcular como se expresa en (2.3). Si se requiere calcular valores sucesivos de la serie, es ineficiente calcular el promedio para cada período, siendo conveniente utilizar la fórmula iterativa expresada en (2.4).

$$SMA_t(x,n) = \frac{x_t + x_{t-1} + \cdots + x_{t-(n-1)}}{n} \quad \textbf{(2.3)}$$

$$SMA_t(x,n) = SMA_{t-1}(x,n) + \frac{x_t - x_{t-n}}{n} \quad \textbf{(2.4)}$$

El período de suavizado $\boldsymbol{n}$ depende del interés del agente en cuanto al tipo de tendencia a capturar. La utilización de $\boldsymbol{n}$ pequeño captura la tendencia a corto plazo, por el contrario, aumentar $\boldsymbol{n}$ es equivalente a considerar más valores del pasado, pudiendo así capturar la tendencia del activo a mediano o largo plazo.

2.2.3.1.2. Media móvil exponencial

La media móvil exponencial (*Exponential Moving Average* <EMA>) corresponde a un filtro de respuesta infinita al impulso [20] (Infinite Impulse Response <IIR>), es decir, en su cálculo utiliza todos los valores pasados de la serie.

En el cálculo de la EMA se ponderan los valores de forma exponencialmente decreciente mientras los términos son más antiguos. Su fórmula recursiva se expresa en (2.5), donde la constante $\alpha \in [0,1]$ es un parámetro que representa la ponderación para cada período de la serie y se calcula como se muestra en (2.6) [18].

$$\begin{aligned} EMA_1 &= x_1 \\ EMA_t(x,n) &= \alpha \cdot x_t + (1-\alpha) \cdot EMA_{t-1}(x,n) \qquad t > 1 \end{aligned} \quad \textbf{(2.5)}$$

$$\alpha = \frac{2}{n+1} \quad \textbf{(2.6)}$$

2.2.3.2. Medias móviles convergentes/divergentes

El indicador técnico de medias móviles convergentes/divergentes (*Moving Average Convergence/Divergence* <MACD>) fue creado por Gerald Appel en la década de los 70 [20], es ampliamente utilizado por inversionistas ya que de éste se puede inferir fuerza, dirección y momento del activo en análisis. Hartle [21] destaca que el MACD provee indicaciones anticipativas de

un potencial cambio de tendencia del mercado. Vakkur [22] muestra que éste corresponde a un excelente indicador de tendencia y que, combinado con el precio, provee señales transables, que permiten evitar caídas del mercado.

El indicador MACD se representa mediante tres señales extraídas de la serie temporal de precio de cierre de un mercado. Estas líneas corresponden a la línea MACD (también llamado MACD), la línea de señal y la diferencia entre ambos, generalmente llamado "histograma".

Para el cálculo del indicador se deben seleccionar tres parámetros: EMA rápida, EMA lenta y MACD EMA. El indicador se calcula como se expresa en (2.7), (2.8) y (2.9) para el MACD principal, señal e histograma respectivamente.

$$MACD_t = EMA_t(\text{Close}, \text{EMA rápida}) - EMA_t(\text{Close}, \text{EMA lenta}) \quad \textbf{(2.7)}$$

$$Signal_t = EMA(MACD_t, \text{MACD EMA}) \quad \textbf{(2.8)}$$

$$Histogram_t = MACD_t - Signal_t \quad \textbf{(2.9)}$$

En la ecuación (2.7) la utilización del concepto de EMA rápida y EMA lenta tiene relación con el tipo de "momento" que se desea extraer. Si la relación entre ambas es grande (Ej: $\text{EMA rápida} = 5$ y $\text{EMA lenta} = 35$), la línea MACD representa el "momento" a corto plazo, por el contrario, si ambos valores son muy parecidos, se extrae el "momento" a largo plazo (Ej: $\text{EMA rápida} = 12$ y $\text{EMA lenta} = 26$). Los conceptos de corto y largo plazo están directamente relacionados con la magnitud de ambos valores seleccionados, por ejemplo utilizar valores de EMA rápida y lenta de doce y 26 respectivamente, puede tener un comportamiento similar a utilizar valores de 40 y 90, sin embargo, el "momento" extraído de estos últimos valores es de mayor plazo.

El indicador MACD provee cuatro tipos de señales transables, estas corresponden a:

- Cruces:
  - La línea MACD cruza la línea de señal (equivalente a que histograma cruce por cero).
  - La línea MACD cruza cero (equivalente a cruce entre EMA rápida y EMA lenta).
- Divergencias:
  - Divergencia de la línea MACD o de señal.
  - Divergencias en el histograma.

El cruce por cero de la línea MACD equivale a tranzar un cruce de medias móviles. Lebaron [23] muestra que estrategias basadas en medias móviles poseen poder predictivo tanto en media como varianza, por lo que la utilización de éstos como variables independientes agrega poder predictivo a un sistema de inversiones.

En la Ilustración 2.4 se muestra el MACD aplicado a ENTEL, en ésta se resaltan patrones divergentes regulares alcistas y ocultos bajistas, mostrando claramente el poder de anticipación de cambios de tendencia de éstas. Por otra parte, se resaltan los cruces entre la EMA rápida y la EMA lenta, marcando cada vez que la línea MACD cruza por cero (color verde para alza y rojo para baja). Finalmente, se destaca que, a diferencia de otros osciladores, el MACD no se restringe a un rango de valores.

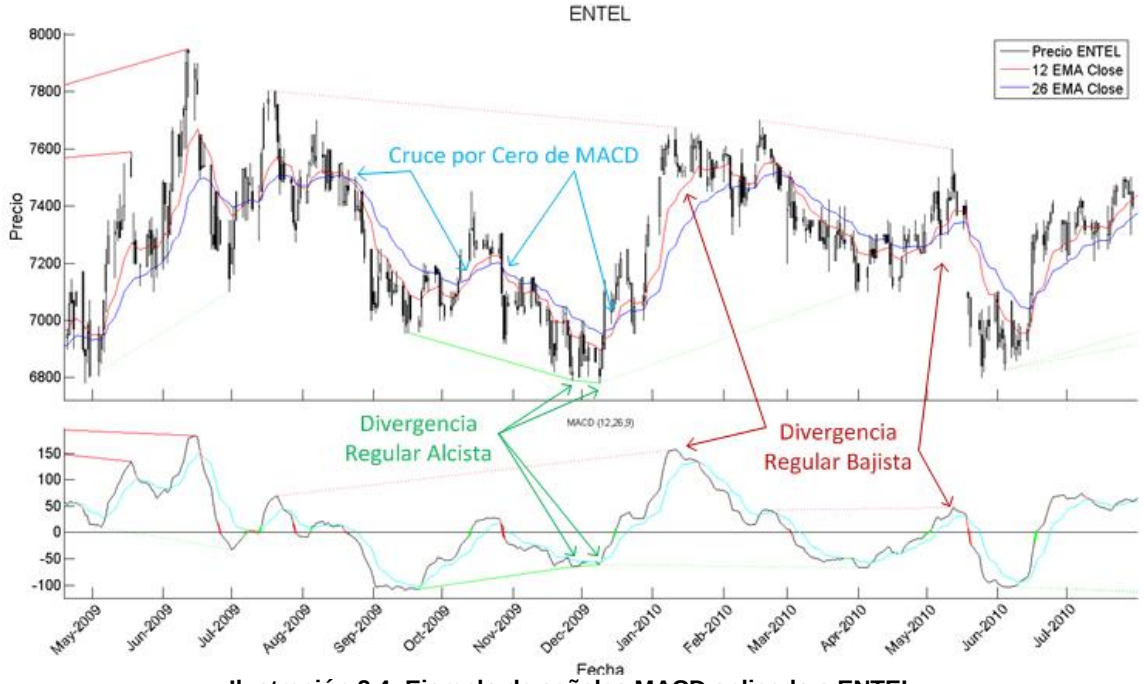

Ilustración 2.4: Ejemplo de señales MACD aplicado a ENTEL

2.2.3.3. Índice de fuerza relativa

El índice de fuerza relativa (*Relative Strength Index* <RSI>) fue ideado y publicado por J. Welles Wilder en 1978 [24], éste corresponde a un indicador técnico del tipo oscilador, es decir, representa la fuerza de la tendencia del activo mediante la comparación de movimientos sucesivos de alza o la baja.

Alfaro y Sagner [25] derivan una expresión del RSI que muestra que éste actúa como un filtro altamente no lineal del precio. Además Alarcón, Pincheira y Selaive [26] exponen que el RSI es ampliamente utilizado en Chile por analistas financieros.

El indicador RSI se calcula normalizando los precios de cierre para los días de alza y baja, como se expone en (2.10) y (2.11) respectivamente. Se suavizan los resultados evaluando la EMA de n períodos y se calcula la relación entre ambos (2.12). Finalmente se normaliza el RSI entre cero y 100 como se expresa en (2.13).

$$U_t = \begin{cases} Close_t - Close_{t-1} & Close_t > Close_{t-1} \\ 0 & \sim \end{cases} \quad \textbf{(2.10)}$$

$$D_t = \begin{cases} Close_{t-1} - Close_t & Close_t < Close_{t-1} \\ 0 & \sim \end{cases} \quad \textbf{(2.11)}$$

$$RS = \frac{EMA(U,n)}{EMA(D,n)} \quad \textbf{(2.12)}$$

$$RSI = 100 - 100 \cdot \frac{1}{1+RS} \quad \textbf{(2.13)}$$

Por definición, en caso de que $EMA(D,n)$ sea cero para un período $\boldsymbol{t}$, el RSI es 100. Del mismo modo si $EMA(U,n)$ es cero, el RSI es cero.

Su interpretación indica niveles de sobrecompra cuando supera el umbral de 70/80 y sobreventa cuando disminuye de 30/20. La utilización de múltiples umbrales se debe a que se suele variar dependiendo de las condiciones de mercado. Por otra parte, generalmente se utiliza un parámetro de $n = 14$.

En la Ilustración 2.5 se muestra el RSI evaluado para ENTEL, en éste se resalta cuando el RSI vuelve de las zonas de sobrecompra y sobreventa con rojo y verde respectivamente.

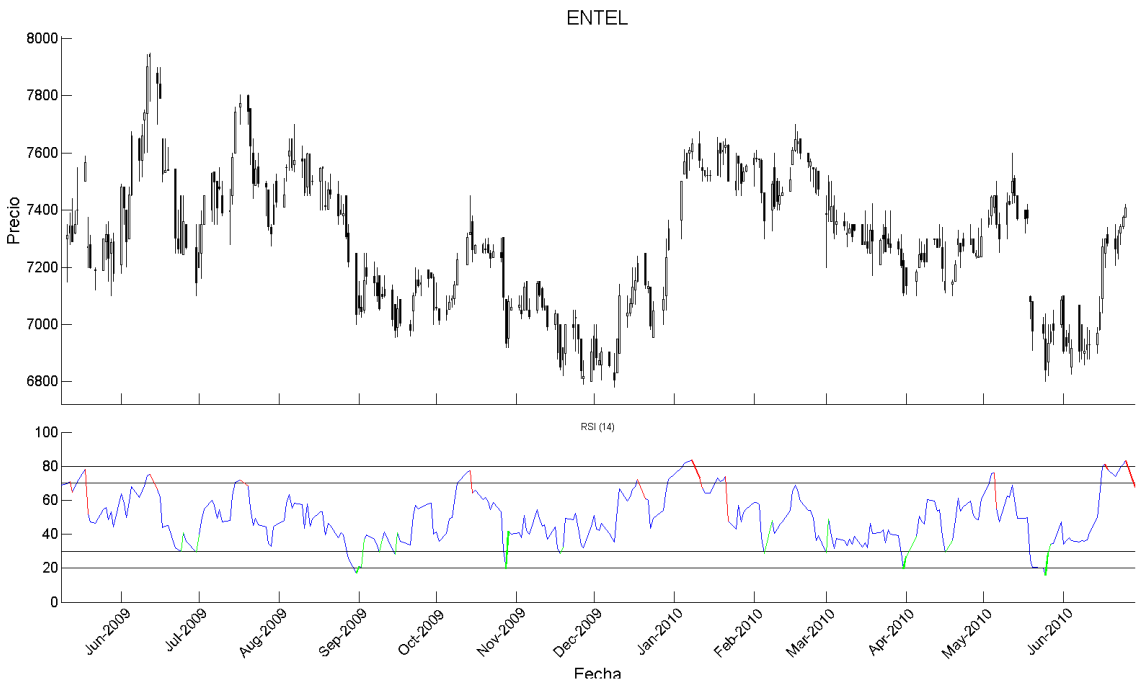

Ilustración 2.5: Ejemplo de señales RSI aplicado a ENTEL

2.2.3.4. Oscilador estocástico

El oscilador estocástico (*Stochastic Oscillator* <*SO*>) corresponde a un indicador de momento que utiliza niveles de soporte y resistencia. Éste indicador intenta anticipar cambio de tendencia del activo financiero, para esto, utiliza los precios de cierre en comparación al último rango de precios. Morris [27] y Luisi [28] destacan el poder predictivo del SO, además, resaltan que éste es capaz de determinar rango y tendencia del activo.

Existen versiones del SO rápido, lento y una versión completa que contiene a ambos [29]. En el cálculo de la versión completa se utilizan tres parámetros: *%K period*, *%D period* y *Slowing*. El SO se conforma por dos series temporales, *%K* corresponde al indicador principal y se calcula como se expresa en (2.14), *%D* representa una versión suavizada del indicador principal y se calcula como se expresa en (2.15).

$$\%K = SMA\left(100 \cdot \frac{Close - L(\%K\ period)}{H(\%K\ period) - L(\%K\ period)}, Slowing\right) \qquad \textbf{(2.14)}$$

$$\%D = SMA(\%K, \%D\ period) \quad \textbf{(2.15)}$$

$$H_t(n) = \max(High_i), \qquad i = t - n + 1, \dots, t \quad \textbf{(2.16)}$$

$$L_t(n) = \min(Low_i), \qquad i = t - n + 1, \dots, t \quad \textbf{(2.17)}$$

Generalmente se utilizan los parámetros $\%K\ period = 5$, $\%D\ period = 3$ y $Slowing = 3$ [30]. Se combina con un indicador de tendencia para identificar pequeños ciclos del mercado. En este indicador, se consideran niveles de sobrecompra con $\%K$ o $\%D$ sobre 70/80 y de sobreventa bajo 30/20. Para identificar ciclos a mediano y largo plazo, se suele utilizar $\%K\ period$ entre 14 y 21.

Del indicador SO se suelen extraer variables de divergencia, agotamiento y cruce, en particular:

- Agotamiento:
  - $\%K$ o $\%D$ se encuentra sobre un umbral de 70/80 para indicar sobrecompra y bajo un umbral de 30/20 para indicar sobreventa.
- Cruces:
  - Entrada y salida de $\%D$ o $\%K$ de un umbral de agotamiento.
  - Cruce entre $\%K$ y $\%D$.
- Divergencia de $\%D$ con respecto al precio cuando la primera oscilación se encuentra en un nivel de agotamiento

En la parte inferior de la Ilustración 2.6 muestra el SO evaluado en ENTEL, en ésta, $\%K$ se muestra en color celeste y $\%D$ en una línea roja punteada. Se observa claramente el poder predictivo de las divergencias sobre este indicador, en especial, el caso a principio de Enero del 2010, donde cuatro divergencias ocurren en el mismo período, con una posterior baja significativa del precio, encontrando una divergencia al alza antes que el precio vuelva a subir.

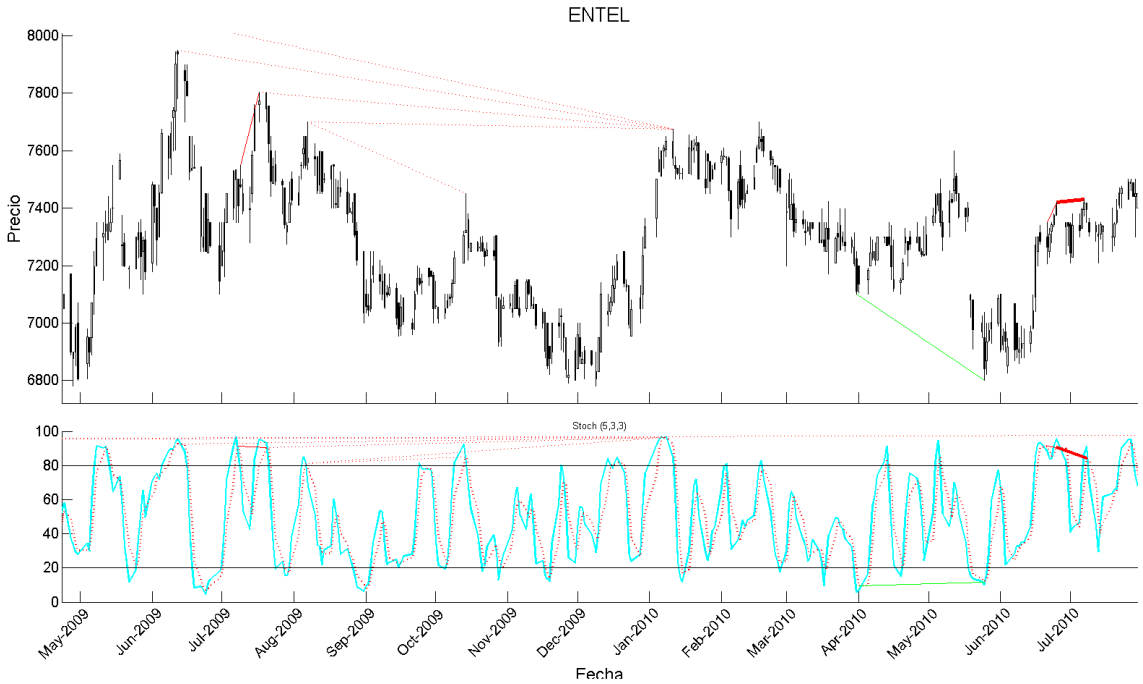

Ilustración 2.6: Ejemplo de señales SO aplicado a ENTEL

2.2.3.5. ZigZag

El ZigZag no corresponde a un indicador de por sí, sin embargo es muy útil como filtro de cambio de precios pequeños de un activo, facilitando el análisis de sólo los movimientos de interés. Por otra parte, se puede combinar con teorías como *Elliot Waves* para encontrar patrones predictivos [31].

Existen diferentes formas de calcular el ZigZag, las que dependen del tipo de precio a ser aplicado y al tipo de mercado. En acciones puede ser aplicado al rango de precio máximo-mínimo (*High-Low)*, siendo más común que sólo se aplique al precio de cierre (*Close*). La definición básica del ZigZag sólo utiliza un parámetro, el cual corresponde a la retracción porcentual mínima del precio antes que se forme una nueva indicación de ZigZag (Anexo B).

Los movimiento de mercado que indica el ZigZag pueden ser clasificados en función de los movimientos anteriores, estos se resumen en “alto más alto” (*Higher High* <HH>) y “alto más bajo” (*Lower High* <LH>) para movimientos de alza, y “bajo más bajo” (*Lower Low* <LL>) y “bajo más alto” (*Higher Low* <HL>) para movimientos de baja (Ilustración 2.7). En general, a los movimientos de alza se les llama *peak* (HH y LH) y a los de baja *trough* (LL y HL).

Para determinar la etiqueta de un ZigZag se utilizan las dos indicaciones previas. Su ecuación se expresa en (2.18), donde ZZ_t representa el t-ésimo período en que se encuentra un ZigZag, con $t > 2$.

$$Label_t = \begin{cases} HH & Close(ZZ_t) > Close(ZZ_{t-1}) \,\&\, Close(ZZ_t) > Close(ZZ_{t-2}) \\ LH & Close(ZZ_t) > Close(ZZ_{t-1}) \,\&\, Close(ZZ_t) < Close(ZZ_{t-2}) \\ LL & Close(ZZ_t) < Close(ZZ_{t-1}) \,\&\, Close(ZZ_t) < Close(ZZ_{t-2}) \\ HL & Close(ZZ_t) < Close(ZZ_{t-1}) \,\&\, Close(ZZ_t) > Close(ZZ_{t-2}) \end{cases} \quad \textbf{(2.18)}$$

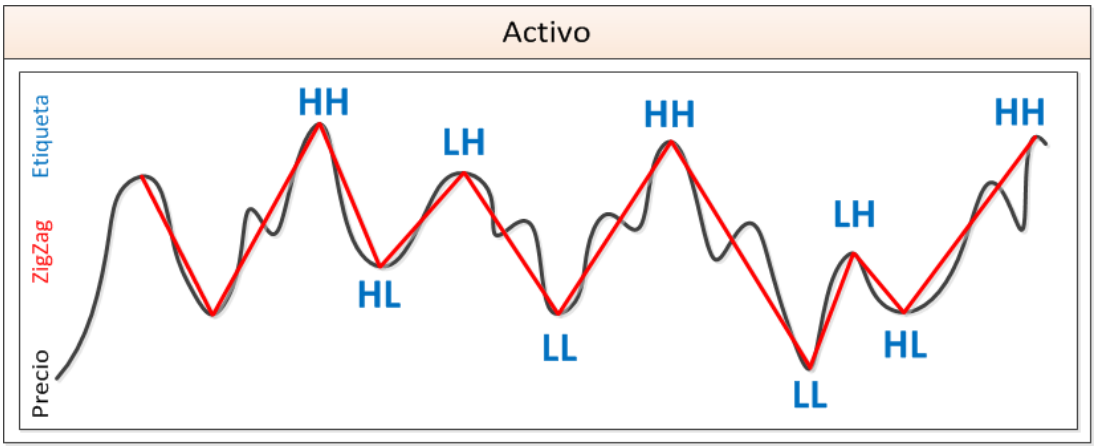

Ilustración 2.7: Ejemplo de ZigZag etiquetado aplicado a precio de activo

De acuerdo a la teoría de Dow [16], el mercado tiene tres tendencias: alcista, bajista y de resistencia. Dow describe una tendencia alcista como un patrón de *peaks* cada vez más altos, es decir, la formación sucesiva de nuevos HH y HL. Por otro lado, una tendencia bajista se describe por la formación sucesiva de nuevos LL y HL.

Cada tendencia posee tres fases, la fase de acumulación representa compra informada por parte de los inversionistas más astutos, la fase de participación pública ocurre cuando el precio del activo tiende a variar rápidamente, aumentan las noticias respecto y los inversionistas que siguen tendencia comienzan a participar. En la fase de distribución, las noticias son mejores que nunca y la participación pública crece, en esta fase los inversionistas informados comienzan a distribuir antes que todos comiencen a vender.

De lo anterior se puede inferir que, una estrategia de inversiones adecuada, requiere la correcta identificación de las fases de acumulación y distribución, comprando y vendiendo activos en el momento apropiado. Lo que es equivalente a adquirir activos en mínimos locales y enajenarlos en máximo locales de la serie temporal, los cuales pueden ser identificados (en el pasado) utilizado el indicador ZigZag.

2.2.3.6. Divergencias

Las divergencias no corresponden a un indicador, sino más bien a un tipo de variables predictivas o atributos extraídos de un indicador. Éstas anticipan cambio de tendencia (debilidad) o continuación de ésta (fortalecimiento).

En un mercado en alcista, si el precio alcanza un HH y el indicador falla en alcanzar un nuevo alto, alcanzando así un LH, se crea una divergencia regular bajista. Por el contario, si el indicador muestra un HH y el precio falla en alcanzar un nuevo HH, teniendo así un LH, se obtiene una divergencia oculta bajista. El mismo concepto se aplica para un mercado bajista.

En la Ilustración 2.8 se muestran las cuatro combinaciones básicas de divergencia, en las columnas se ordenan según patrón divergente (regular u oculto) y en las filas según tipo de tendencia (alcista o bajista). En cada cuadrante en la parte superior se grafica la evolución temporal del precio del activo y en la inferior el indicador con el que se forma el patrón divergente.

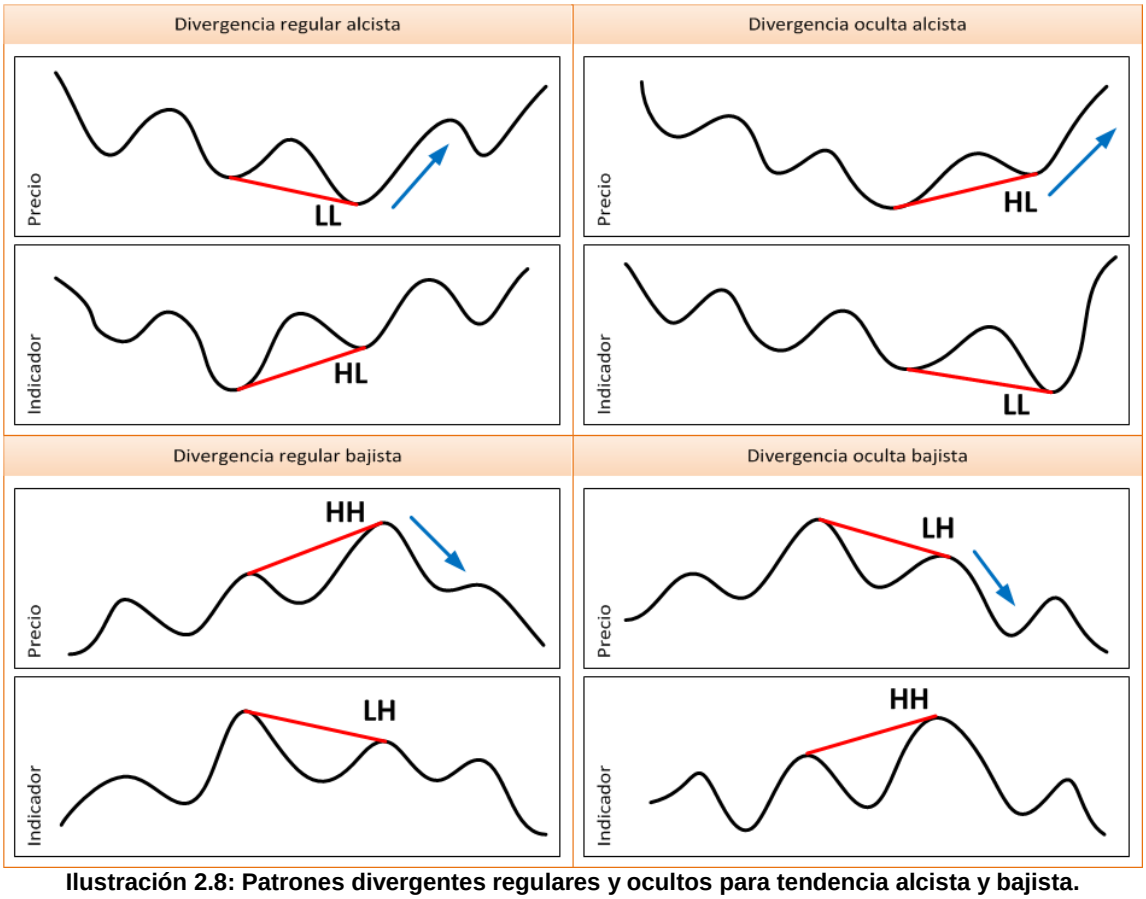

Ilustración 2.8: Patrones divergentes regulares y ocultos para tendencia alcista y bajista.

Se infiere que se pueden encontrar divergencias en múltiples períodos de tiempo, por lo que las diferencias temporales entre el *peak* o *trough* de inicio y fin de divergencia varía. Las divergencias regulares anticipan cambio de tendencia, por otra parte, las divergencias ocultas anuncian una posible continuación de ésta.

Existe un caso especial de divergencia, éste consiste en una divergencia que comienza en el período en que termina otra divergencia (divergencias continuadas). A este tipo de indicador de cambio de tendencia se suele llamar divergencia triple, ya que que utiliza tres *peaks* o *troughs* de precio y de indicador (Ilustración 2.9). Con el fin de mantener la concordancia en el espacio temporal, se suele conceder validez sólo a divergencias triples con diferencia temporal similar entre sus dos divergencias simples.

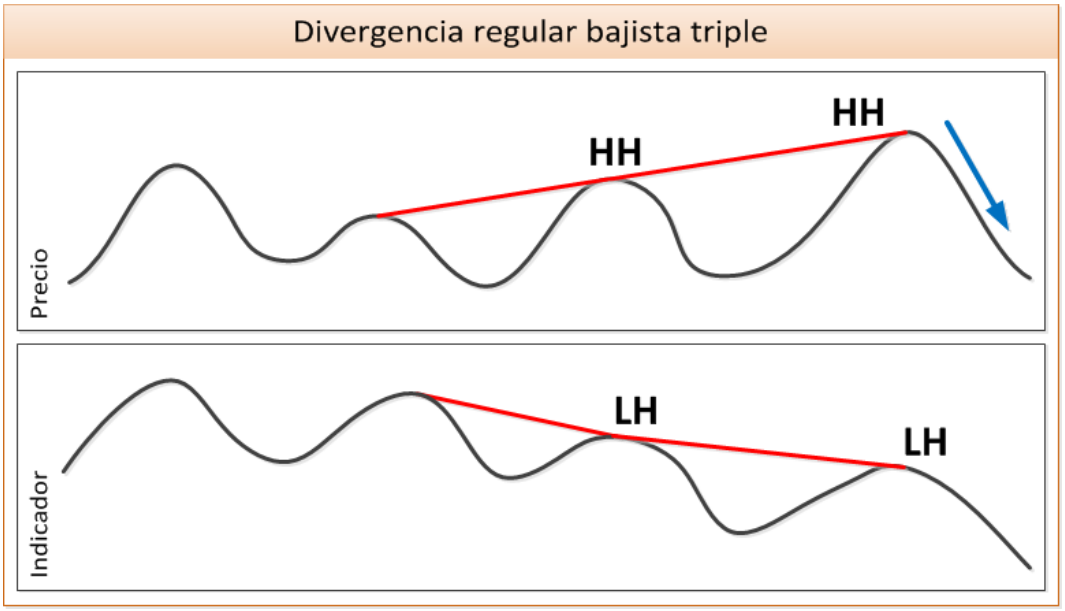

Ilustración 2.9: Patrón de divergencia regular bajista triple

2.3. Proceso de extracción de conocimiento

El proceso de Extracción de conocimiento (*Knowledge Discovery from Databases* <KDD>) se refiere al proceso no trivial de descubrir potencial conocimiento en bases de datos. Éste corresponde a un proceso exhaustivo e iterativo que explora grandes volúmenes de datos para determinar relaciones. Fayyad [1] destaca que el arte y habilidad de extraer conocimiento aún no es superado por una máquina, sin embargo, los grandes volúmenes de datos con los que se trabaja hacen de éstas una herramienta indispensable.

En la Ilustración 2.10 se esquematizan las cinco fases del proceso KDD, estas corresponden a:

1. **Selección de datos:** Se determina el tipo de información a utilizar y las fuentes de datos.

2. **Pre-procesamiento:** Consiste en la preparación de los datos extraídos en la etapa anterior, en esta etapa se utilizan distintas estrategias para manejar datos inconsistentes o fuera de rango (*outliers*) y datos faltantes o blancos. Su objetivo es obtener la estructura adecuada de los datos para su posterior transformación.

3. **Transformación:** Se realiza el tratamiento preliminar de los datos, se transforman y generan nuevas variables. Consolidando los datos de forma apropiada para la correcta utilización de los modelos en la fase de minería de datos.

4. **Minería de datos:** Consiste en el modelamiento de los datos ya transformados, donde métodos "inteligentes" son aplicados con el objetivo de extraer patrones desconocidos y potencialmente útiles que se encontraban contenidos en ellos.

5. **Interpretación y evaluación:** En la fase final se identifican los patrones encontrados que sean de interés y se realiza una evaluación de los resultados obtenidos.

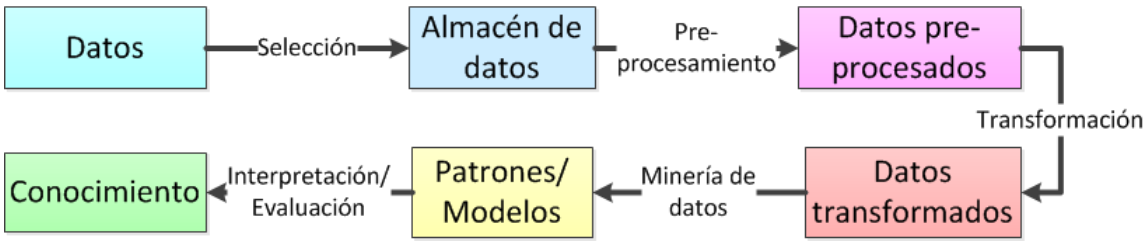

Ilustración 2.10: Proceso KDD

2.4. Minería de datos

Minería de datos corresponde a un área de las ciencias de la computación y las matemáticas que utiliza distintos métodos o algoritmos para descubrir patrones en el conjunto de datos.

Dentro de minería de datos existen tres grandes paradigmas: Aprendizaje supervisado, no supervisado y retroalimentado. La diferencia principal entre los dos primeros yace en que aprendizaje supervisado utiliza datos con una variable objetivo, es decir, utiliza un conjunto de datos con información de antemano, diferenciando atributos que se desea sean relacionados a una variable dependiente. Por otra parte, aprendizaje no supervisado intenta encontrar estructura o patrones en los datos sin conocimiento previo.

En ésta fase se suele trabajar con la base de datos dividida en tres segmentos: Entrenamiento; validación y testeo. La base de entrenamiento se utiliza para que el algoritmo o modelo paramétrico ajuste sus parámetros internos (Ej: BPNN ajuste los pesos sinápticos). Por otra parte, la base de validación se utiliza para buscar los mejores parámetros del modelo utilizado (Ej: Número de neuronas en la capa oculta de una red MLP). Es decir, se utiliza la base de entrenamiento para que un modelo paramétrico se adapte probando distintas combinaciones de parámetros, luego este modelo predice sobre la base de validación, escogiéndose la combinación de parámetros que dan mejor resultado (mediante alguna métrica). De esta manera, se evita sobre o bajo ajuste del modelo utilizado, maximizando su poder predictivo.

2.4.1. Kernel

En el contexto de aprendizaje de máquina, la utilización del método o "truco" de kernel fue publicado por primera vez en 1964 [32]. Éste consiste en el mapeo de datos desde un set genérico S, hacia un espacio de producto interno V, siendo así, una clasificación lineal en V, equivalente a una clasificación no lineal en S.

Este método se utiliza en algoritmos de aprendizaje de máquinas que sólo requieran producto punto entre vectores en V, eligiendo un mapeo tal que, los productos puntos de alta dimensión, se puedan calcular en el espacio original a través de funciones de kernel. Formalmente, una función $K: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ es llamado kernel en $\mathbb{R}^d$, si existe un función $\phi: \mathbb{R}^d \to \mathcal{F}$ en algún espacio $\mathcal{F}$ con producto escalar $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ que cumpla (2.19) para todo $x, y \in \mathbb{R}^d$.

$$K(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{F}} \qquad \textbf{(2.19)}$$

Entre las funciones de kernel más comunes se encuentran: Kernel Polinomial homogéneo (2.20) y no homogéneo (2.21); función Gaussiana (2.22) y tangente hiperbólica (2.23). En la Ilustración 2.11 se ejemplifica la función de kernel Gaussiano para distintos anchos de kernel.

$$K(x,y) = (x \cdot y)^d, \qquad d > 0 \tag{2.20}$$

$$K(x,y) = (x \cdot y + 1)^d, \qquad d > 0 \tag{2.21}$$

$$K(x,y) = e^{-\gamma \|x-y\|^2}, \qquad \gamma > 0 \tag{2.22}$$

$$K(x,y) = tanh(\kappa x \cdot y + c), \qquad \kappa > 0, c > 0 \tag{2.23}$$

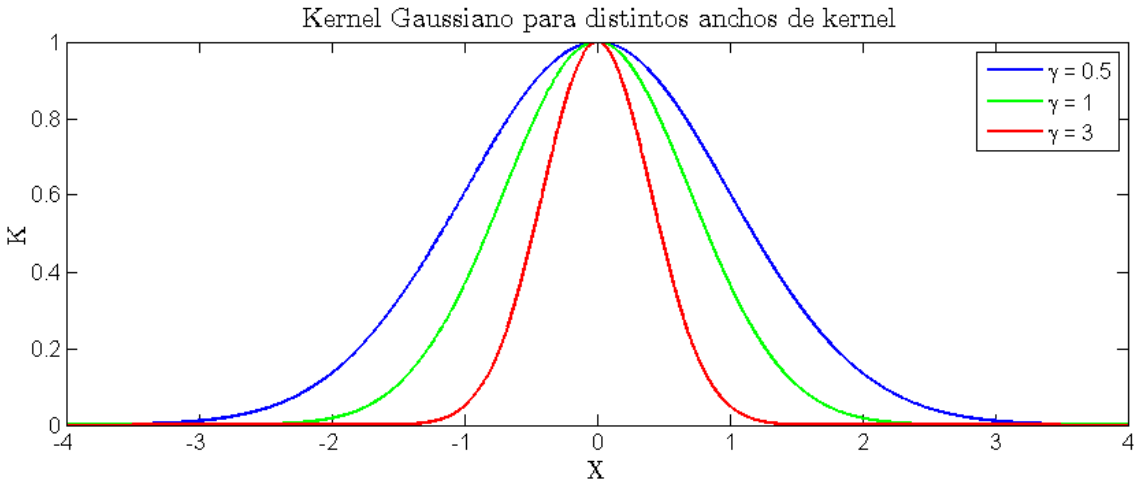

Ilustración 2.11: Kernel Gaussiano con distintos anchos de kernel γ

En la Ilustración 2.12 se aprecia un plano observaciones pertenecientes a dos clases. En el espacio original, los atributos no pueden ser separados linealmente, sin embargo, al aplicar un kernel, éstos se pueden separar linealmente por un plano (recta en dos dimensiones) en el espacio de producto interno.

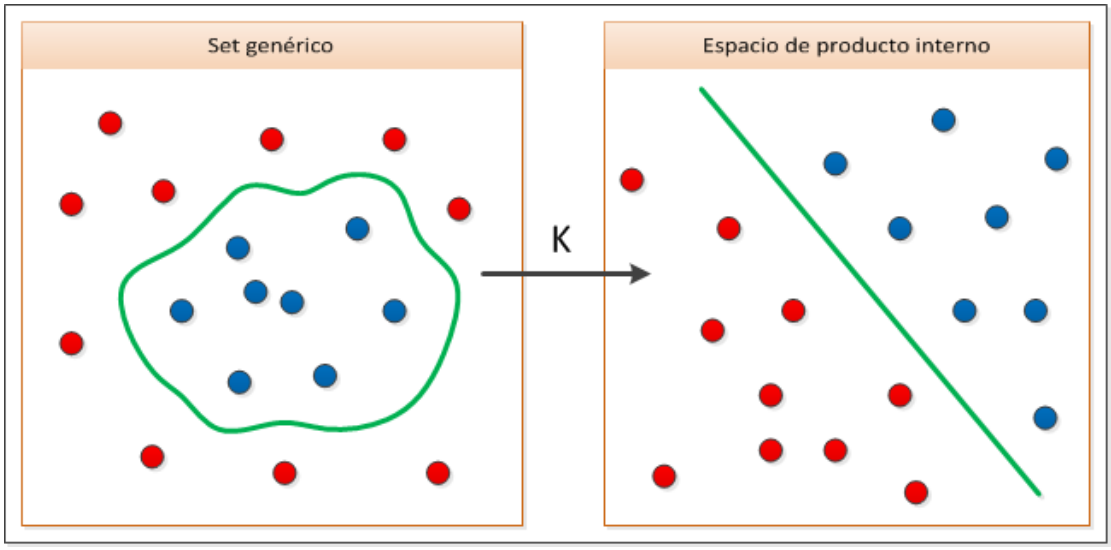

Ilustración 2.12: Ejemplo de aplicación de kernel sobre set genérico

2.4.2. Selección de características

Selección de características puede ser considerada parte de la fase de pre-procesamiento o de minería de datos, su objetivo es encontrar el subconjunto de atributos con mayor valor predictivo, evitando así utilizar variables que agreguen ruido en la fase de entrenamiento, mejorando la predicción y acelerando el proceso de adaptación de los modelos. Entre los enfoques más utilizados se encuentran:

- *Forward Feature Selection* (FS): Se comienza sin atributos en el modelo, se agregan de a una las variables y se evalúa bajo cierta métrica el desempeño de agregar esta variable, eligiéndose, de ellas, la que mejore más el desempeño (si es que se mejora). El proceso se repite hasta que ninguna variable mejore el rendimiento al ser agregada.

- *Backward Elimination* (BE)*:* En este enfoque se comienza con todos los atributos, luego se evalúa la eliminación de cada variable, eliminándose efectivamente la variable con mayor aumento de desempeño al ser eliminada (si es que alguna lo mejora). El proceso se repite hasta que ninguna mejora sea posible.

2.4.3. Aprendizaje supervisado

Aprendizaje supervisado corresponde a un paradigma de aprendizaje de máquina vastamente estudiado los últimos años. En aprendizaje supervisado, es necesario un conjunto de pares de ejemplo $(x, y), x \in X, y \in Y$. Su objetivo es encontrar una función $f: X \longrightarrow Y$ que calce o mapee los ejemplos. Se asume que los datos implícitamente contienen conocimiento previo sobre el dominio del problema.

El aprendizaje supervisado se utiliza esencialmente para realizar regresiones o clasificaciones. Dentro de los algoritmos más utilizados se encuentran Redes Neuronales Artificiales de Retropropagación (BPNN) y Máquinas de Soporte Vectorial (SVM), debido a su gran capacidad de extracción de conocimiento no lineal entre las variables. Además, se destaca Modelos Basados en Similitud (SBM), debido a su capacidad de ajustarse a los datos, regresión lineal debido a su fácil interpretación y los clasificadores Bayesianos (NB) debido a su bajo costo computacional.

2.4.3.1. Regresión lineal

La regresión lineal es un modelo matemático que relaciona las variables dependientes e independientes de forma lineal (puede ser aplicado un kernel previamente). En términos estadísticos, un modelo de regresión lineal "simple" es el estimador lineal de mínimos cuadrados para una variable dependiente y "multivariado" para varias de éstas. Corresponde al primer tipo de análisis de regresión rigurosamente estudiado y ampliamente utilizado en la práctica.

El problema de optimización para un modelo se regresión simple utilizando mínimos cuadrados se expresa en (2.24). En éste, el término cuadrático representa el error entre la variable predictiva x y la variable objetivo y para cada término $\boldsymbol{i}$ de las $\boldsymbol{m}$ muestras. Expandiendo los términos para encontrar estimaciones de α y β, se obtienen los valores que minimizan la función objetivo como se expresa en (2.25) y (2.26), en éstas, $\bar{x}$ e $\bar{y}$ representan la media de la variables predictivas y dependiente respectivamente. Finalmente, el modelo predictivo para un nuevo valor x_n se expresa en (2.27).

$$\min_{\alpha,\beta} \sum_{i=1}^{m} (y_i - \alpha - \beta \cdot x_i)^2 \quad \textbf{(2.24)}$$

$$\hat{\beta} = \frac{\sum_{i=1}^{m} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{i=1}^{m} (x_i - \bar{x})^2} \quad \textbf{(2.25)}$$

$$\hat{\alpha} = \bar{y} - \hat{\beta} \cdot \bar{x} \quad \textbf{(2.26)}$$

$$y_n = \hat{\alpha} + \hat{\beta} \cdot x_n \quad \textbf{(2.27)}$$

En la Ilustración 2.13 se ejemplifica un ajuste mediante regresión lineal simple para un conjunto de datos (X, Y) en dos dimensiones. En ésta se aprecia que, en términos simples, la regresión lineal corresponde al ajuste de una recta entre la variable independiente y objetivo, siendo α y β el sesgo y la pendiente de la recta respectivamente.

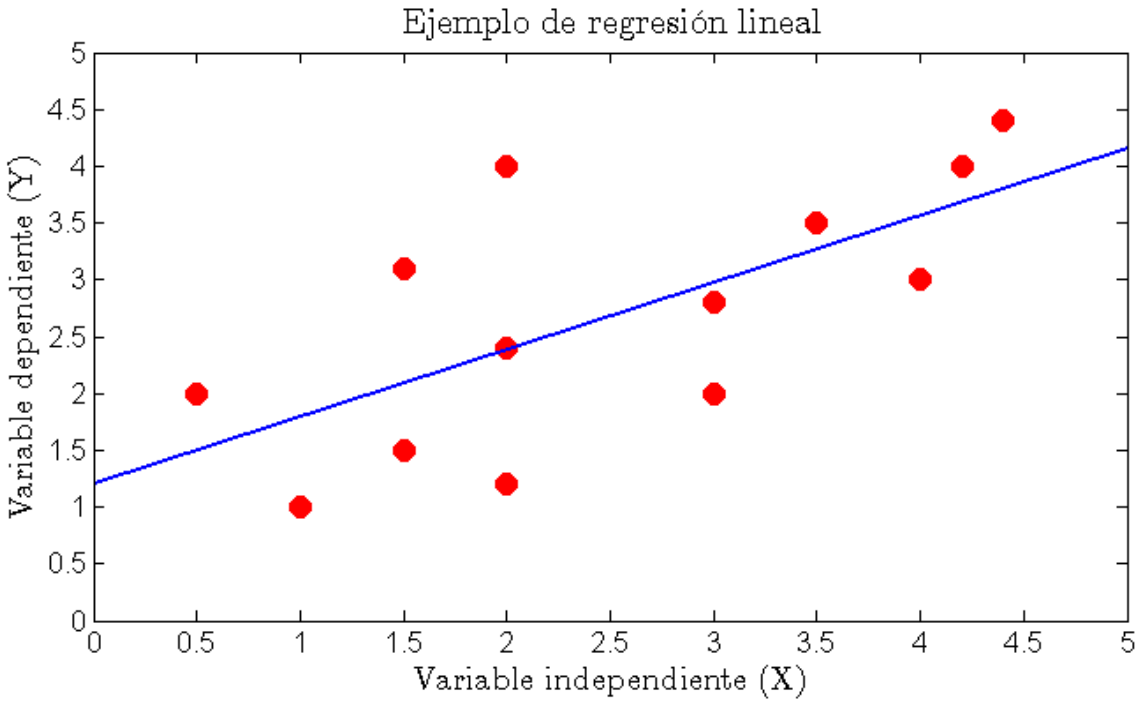

Ilustración 2.13: Ejemplo de regresión lineal

2.4.3.2. Clasificador Bayesiano

Las redes Bayesianas son un algoritmo probabilístico de aprendizaje supervisado. Éstos consideran que los atributos contribuyen de forma independiente en la predicción. En la mayoría de las aplicaciones, la estimación de los parámetros del modelo de clasificador Bayesiano se realiza por máxima verosimilitud. Son vastamente utilizados debido a su fácil interpretación y bajo costo computacional, el que se atribuye al supuesto de independencia de las variables independientes, por lo que sólo se deben calcular las medias y varianzas para cada variable por separado y no la matriz de covarianzas completa.

El clasificador Bayesiano corresponde a la forma más simple de las Redes Bayesianas. En éste, ningún atributo posee padre, excepto la variable dependiente u objetivo. Estas relaciones se pueden apreciar en la Ilustración 2.14.

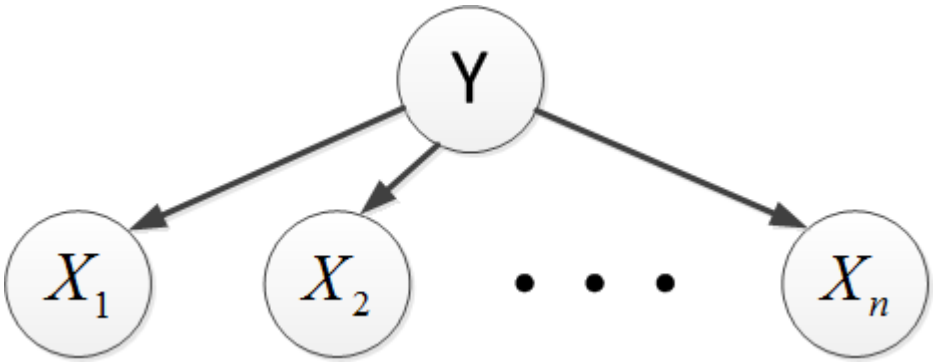

Ilustración 2.14: Diagramas de dependencias en clasificador Bayesiano ingenuo

Si se poseen las clases $y \in \{0,1\}$ y una muestra $X = \{x_1, \cdots, x_n\}$ con $\boldsymbol{n}$ atributos. Utilizando teorema de Bayes, la probabilidad de que X pertenezca a la clase y se expresa en (2.28). Por lo tanto, X es clasificado en la clase $y = 1$ sólo si se cumple (2.29), donde $f_b(X)$ es llamado el clasificador de Bayes. Si se asume que las variables son independientes entre sí, entonces se cumple (2.30), por lo que el clasificador Bayesiano ingenuo (*Naive Bayes Classifier* <NB>) se puede formular como se expresa en (2.31).

$$p(y|X) = \frac{p(X|y) \cdot p(y)}{p(X)} \quad \textbf{(2.28)}$$

$$f_b(X) = \frac{p(y = 1|X)}{p(y = 0|X)} \geq 1 \quad \textbf{(2.29)}$$

$$p(X|y) = p(x_1, \cdots, x_n|y) = \prod_{i=1}^{n} p(x_i|y) \quad \textbf{(2.30)}$$

$$f_{nb}(X) = \frac{p(y = 1)}{p(y = 0)} \cdot \prod_{i=1}^{n} \frac{p(x_i|y = 1)}{p(x_i|y = 0)} \quad \textbf{(2.31)}$$

2.4.3.3. Red neuronal artificial de retropropagación

Las redes neuronales artificiales son un paradigma de aprendizaje de máquina inspirado en el funcionamiento del sistema nervioso animal. Éstas corresponden a una estructura de elementos de procesamiento altamente

conectados, los cuales ajustan sus interconexiones para resolver problemas en específico.

El elemento más básico de una red neuronal es llamado “neurona” o “nodo”. Éste elemento toma las entradas, las pondera por los pesos sinápticos y suma entre ellas, entregando el resultado como argumento a una función de activación. Matemáticamente se expresa en (2.32) y puede ser esquematizado como se expone en la Ilustración 2.15.

$$y_k = g(z) = g\left(\sum_{i=1}^{n} \theta_{k,i} \cdot x_i + \theta_{k,0}\right) \qquad \textbf{(2.32)}$$

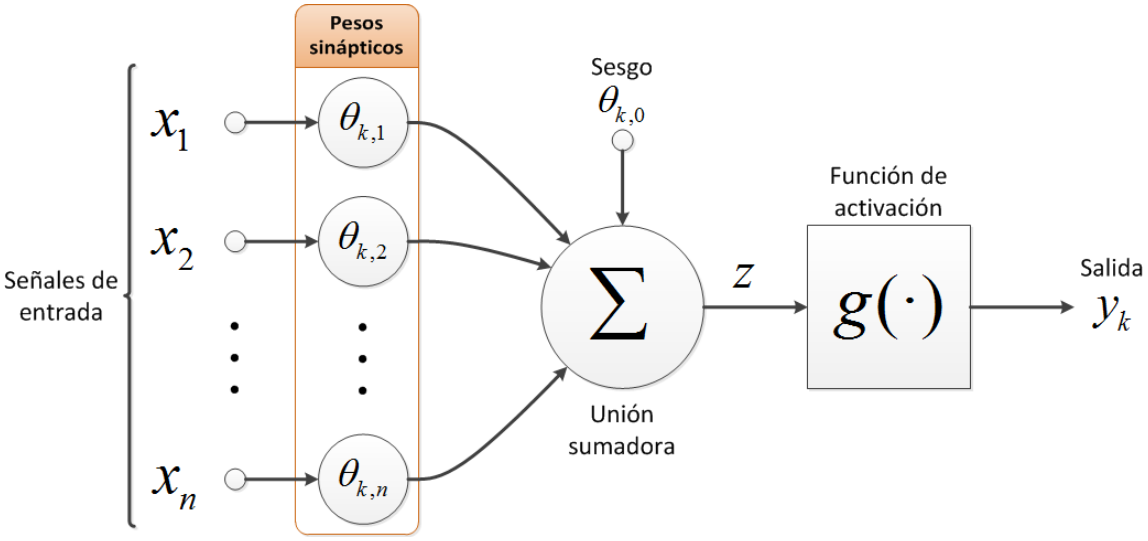

Ilustración 2.15: Diagrama de una neurona

Las entradas y salidas de una neurona pueden ser respectivamente salidas y entradas de otra neurona, formando así una red neuronal. En general, la función de activación puede ser cualquiera, sin embargo, las más frecuentes son la función sigmoide (2.33) y tangente hiperbólica (2.34).

$$y_k = g(z) = \frac{1}{1 + e^{-z}}, \qquad y_k \in [0,1] \qquad \textbf{(2.33)}$$

$$y_k = g(z) = \tanh(z), \qquad y_k \in [-1,1] \qquad \textbf{(2.34)}$$

La combinación de neuronas más utilizada y aplicada a problemas reales es el perceptrón multicapa (*Multi Layer Perceptron* <MLP>), también conocida como red *feedforward*. La red MLP básica se conforma ordenando las neuronas en capas, luego las neuronas de cada capa tienen como entrada las salidas de las neuronas de la capa anterior.

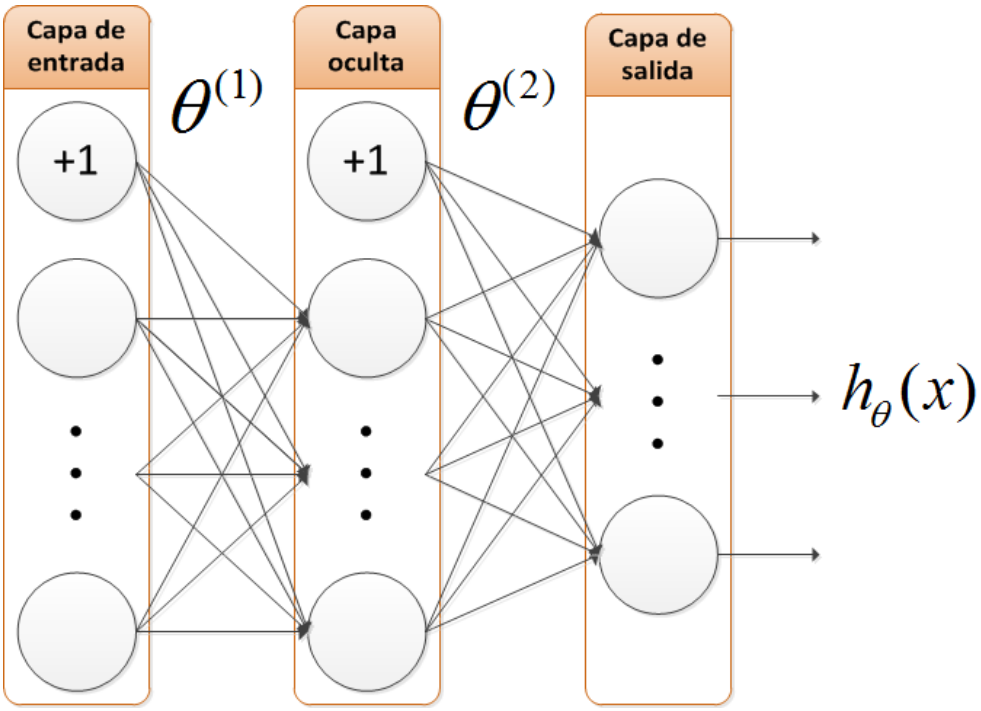

Ilustración 2.16: Modelo de red neuronal (MLP)

En la Ilustración 2.16 se aprecia una red MLP con una capa de entrada y dos capas de neuronas. La primera capa de neuronas se suele llamar “capa oculta” (*Hidden Layer*) y la segunda es llamada “capa de salida” (*Output Layer*). El término “oculta” se refiere a que no es visible como las capas de entrada y salida. Se pueden tener tantas como se desee, aumentando las interconexiones o no-linealidad del sistema. Así, el modelamiento de una Red Neuronal con configuración MLP permite expresar sus componentes matricialmente, permitiendo la definición de los pesos en cada capa como $\theta^{(l)},\ l \in [1, \#Capas]$.

En el entrenamiento de la BPNN se destacan dos procesos, propagación o *feedforward* y retropropagación o *backpropagation*. Dichos procesos predicen y ajustan los pesos sinápticos de la red respectivamente. Si se define la matriz de $\boldsymbol{m}$ ejemplos de entrada como se expresa en (2.35) y de salida en (2.36), luego el proceso de *feedforward* de la red MLP se inicializa como se expresa en (2.37). Se procede iterando para todas las capas utilizando (2.38) y (2.39) para el proceso de propagación. El término $h_\theta(x)$ en (2.40) representa la salida estimada por la red MLP tras aplicar los pesos θ para una entrada cualquiera x.

$$X = \begin{bmatrix} -\left(x^{(1)}\right)^T - \\ -\left(x^{(2)}\right)^T - \\ \vdots \\ -\left(x^{(m)}\right)^T - \end{bmatrix} \tag{2.35}$$

$$Y = \begin{bmatrix} -\left(y^{(1)}\right)^T - \\ -\left(y^{(2)}\right)^T - \\ \vdots \\ -\left(y^{(m)}\right)^T - \end{bmatrix}, \qquad y^{(t)} = \begin{bmatrix} v_1^{(t)} \\ v_2^{(t)} \\ \vdots \\ v_{\#Salidas}^{(t)} \end{bmatrix}, \qquad v_k^{(t)} = \begin{cases} 1 & si\ y^{(t)} \in Clase\ k \\ 0 & \sim \end{cases} \tag{2.36}$$

$$a^{(1)} = [1 \quad x] \tag{2.37}$$

$$z^{(l)} = \theta^{(l-1)} a^{(l-1)}, \qquad l = 2, \cdots, \#Capas \tag{2.38}$$

$$a^{(l)} = g([1 \quad z^{(l)}]), \qquad l = 2, \cdots, \#Capas \tag{2.39}$$

$$h_\theta(x) = a^{(\#Capas)} \tag{2.40}$$

El proceso de retropropagación se inicia tras la obtención de la hipótesis $h_\theta(x)$ en el proceso de propagación. En éste se debe calcular un término de "error" $\delta_k^{(l)}$ a cada nodo $\boldsymbol{k}$ en la capa $\boldsymbol{l}$, que representa qué tan "sensitivo" ha sido ese nodo para errores en la salida. Para cada nodo de salida se puede medir directamente la diferencia con el valor objetivo verdadero y utilizar dichos valores para definir $\delta_k^{(\#Capas)}$, como se expresa para cada ejemplo $\boldsymbol{j}$ en (2.41).

Para las capas ocultas se puede calcular $\delta_j^{(l)}$ utilizando los términos de error de la capa $(l+1)$, tal como se expresa en la ecuación (2.42). En ésta el operador $(.*)$ representa la multiplicación casilla a casilla de las matrices involucradas. Por otra parte, la función $g'(\cdot)$ corresponde a la derivada de la función de activación $g(\cdot)$, para el caso de la función sigmoide expresada en (2.33); su derivada se presenta en (2.43).

$$\delta_j^{(\#Capas)} = J\left(a_j^{(\#Capas)} - y_j\right), \qquad j = 1, \cdots, m \tag{2.41}$$

$$\delta^{(l)} = \left(\theta^{(l)}\right)^T \delta^{(l+1)} .* g'\left(z^{(l)}\right), \qquad l = 2, \cdots, \#Capas - 1 \tag{2.42}$$

$$g'(z) = \frac{d}{dz} g(z) = g(z)(1 - g(z)) \tag{2.43}$$

La función de costo $\boldsymbol{J}$ de la ecuación (2.41) puede variar; generalmente se utiliza como error los mínimos cuadrados (*Minimum Square Error* <MSE>).

La función utilizada en esta memoria se expresa en (2.44), donde los términos superiores corresponden al error de la hipótesis $\left(h_\theta\left(x^{(i)}\right)\right)_k$ respecto al valor real $y_k^{(i)}$. La parte inferior regulariza los pesos para evitar sobreajuste, añadiendo un costo mientras mayores sean éstos. El parámetro λ representa cuánta importancia se le da a la regularización y, en conjunto con el número de neuronas y capas ocultas, debe ser optimizado.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[-y_k^{(i)} log\left(\left(h_\theta\left(x^{(i)}\right)\right)_k\right) - (1-y_k^{(i)})\log(1-\left(h_\theta\left(x^{(i)}\right)\right)_k)\right] + \frac{\lambda}{2m}\sum_{l=1}^{\#Capas}\sum_{j=1}^{\#salidasCapa_l}\sum_{k=1}^{\#entradasCapa_l}\left(\theta_{j,k}^{(l)}\right)^2 \quad \textbf{(2.44)}$$

La predicción final para una observación se realiza tomando el índice de la neurona de la capa de salida con mayor argumento.

La utilización efectiva de una red neuronal implica múltiples entrenamientos para distintos valores de neuronas en cada capa oculta, número de capas ocultas y factor de regularización λ. A causa del alto costo computacional que implica probar todas las combinaciones, un enfoque tradicional es fijar el número de capas ocultas, utilizando el mismo número de neuronas en cada capa. Luego se varía el número de neuronas linealmente y λ exponencialmente en una búsqueda de grilla (Ej: Si se fija una capa oculta, luego se pueden probar los valores $\#Neuronas \in \{5,10,\cdots,100\}$ y $\lambda \in \{2^{-9}, 2^{-7}, \cdots, 2^{9}\}$).

2.4.3.4. Máquina de soporte vectorial

Las máquinas de soporte vectorial (*Support Vector Machine* <SVM>) son modelos de aprendizaje supervisado utilizados para encontrar patrones en los datos. Éstos buscan un hiperplano que separe de forma óptima las observaciones de dos clases. Los atributos pueden haber sido previamente proyectados a un espacio de dimensionalidad superior a través de algún kernel.

La optimización del hiperplano separador de clases se basa en buscar el hiperplano que maximice el margen o distancia con los puntos más cercanos al mismo (vector de soporte). Por ejemplo, en la Ilustración 2.17 se distinguen dos clases (rojo y azul) en un plano bidimensional. En ésta se observa que tanto H_1 como H_2 son planos separadores, sin embargo, H_2 posee mayor margen

respecto a las observaciones, por lo que éste se catalogaría como un mejor hiperplano separador.

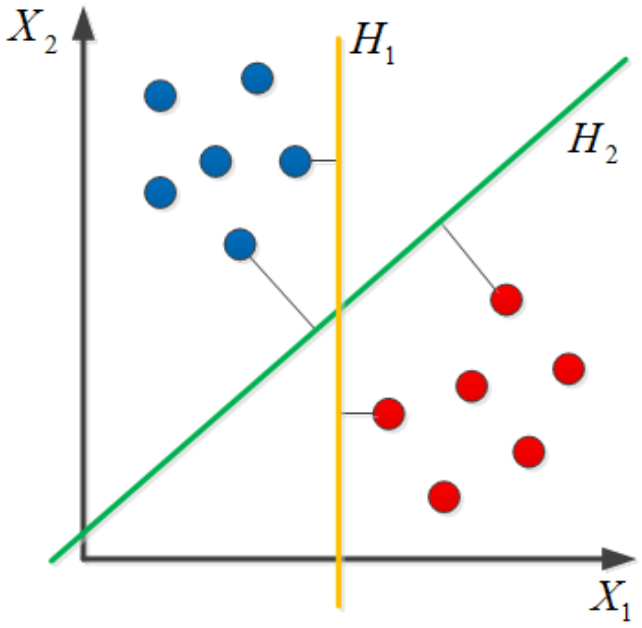

Ilustración 2.17: Ejemplos de planos separadores en dos dimensiones

En 1995 se incluye una modificación de SVM que permite tener ejemplos mal clasificados [33]. Si no existe un hiperplano que pueda separar todos los ejemplos, el método del margen suave (*Soft Margin*) escoge un hiperplano pseudo-óptimo para realizar la clasificación. Esta variante introduce el parámetro ξ_i, que mide el grado de ejemplos mal clasificados de la clase $\boldsymbol{x_i}$, quedando así la condición para cada observación como se expresa en (2.45). En ésta, (x_i, y_i) son el conjunto de variable independiente y dependiente de la observación $\boldsymbol{i}$, con $y_i \in \{-1,1\}$.

$$y_i(\boldsymbol{w} \cdot x_i - b) \geq 1 - \xi_i, \qquad 1 \leq i \leq n, \ \ \xi_i \geq 0 \qquad \textbf{(2.45)}$$

La función objetivo debe incluir una penalización por los valores mal clasificados, convirtiéndose en el problema de optimización expresado en (2.46). Éste representa un balance entre maximizar el rango del hiperplano separador y minimizar los ejemplos mal clasificados. Si se añade la restricción (2.45), el problema se puede solucionar utilizando multiplicadores de Lagrange, lo que se reduce en resolver (2.47), donde $\alpha_i, \beta_i \geq 0$.

$$arg\min_{\boldsymbol{w},\xi,b}\left\{\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i\right\} \quad \textbf{(2.46)}$$

$$arg\min_{\boldsymbol{w},\xi,b}\max_{\alpha,\beta}\left\{\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y_i(w\cdot x_i - b) - 1 + \xi_i] - \sum_{i=1}^{n}\beta_i\xi_i\right\} \quad \textbf{(2.47)}$$

La utilización apropiada de SVM requiere la optimización de los parámetros de margen C y del kernel (si es que se utiliza uno. Para esto, el entrenamiento se debe realizar con distintos valores, seleccionando la combinación de parámetros con mejor rendimiento (en base a una métrica) sobre la base de validación. Una decisión común es la utilización de *kernel* Gaussiano, el cual posee un único parámetro de ancho γ. La mejor combinación de C y γ se realiza típicamente en una búsqueda de grilla, variando ambos parámetros de forma exponencial [34], por ejemplo $C \in \{2^{-5}, 2^{-3}, \cdots, 2^{11}\}$ y $\gamma \in \{2^{-11}, 2^{-9}, \cdots, 2^{3}\}$.

2.4.3.5. Métodos multivariados basados en similitud

Los métodos basados en similitud (*Similarity Based Method* <SBM>) corresponden a un caso específico de análisis de espacios de *kernel* [35] (*Space kernel Analysis* <SKA>), diseñados para ajustarse exactamente a los datos [36]. Wegerich [37] muestra que el modelamiento empírico y no paramétrico que entrega SBM, permite el reconocimiento de patrones para generar estimaciones de valores en bases de datos modeladas. Holtan y Wheeler [38] Nieman y Olson [39] y Wegerich, Wilks y Pipke [40] respaldan esta hipótesis monitoreando satisfactoriamente distintos recursos. Concluyendo que SBM satisface los requerimientos básicos de escalabilidad, flexibilidad y usabilidad.

Si se definen las entradas X_{tr} y salidas objetivo Y_{tr} del sistema, como se expone en (2.35) y (2.36) respectivamente, la forma general de SKA para evaluar un vector de entrada X_v se expresa en (2.48), donde para realizar la evaluación de una matriz de entradas basta tomar la división casilla a casilla.

En (2.48), $\otimes$ es el operador de similitud definido como $(X_{tr}^T \otimes X_n) = [K(X_1, X_v), \cdots, K(X_n, X_v)]^T$, $\mathbf{1}$ es un vector de $m \times 1$ con todos los elementos siendo uno y $K_s(X_{tr}, X_n) = A \cdot (X_{tr}^T \otimes X_n)$ es el kernel de espacio. SMB corresponde a una interpolación de SKA y se expresa en (2.49), ésta equivale a (2.48) reemplazando $A = G^{-1} = (X_{tr}^T \otimes X_v)^{-1}$.

$$Y_v = \hat{f}_{SKA}(X_v) = \frac{Y_{tr}^T \cdot A \cdot (X_{tr}^T \otimes X_v)}{\mathbf{1}^T \cdot A \cdot (X_{tr}^T \otimes X_v)} = \frac{Y_{tr}^T \cdot K_s(X_{tr}, X_v)}{\mathbf{1}^T \cdot K_s(X_{tr}, X_v)} \quad \textbf{(2.48)}$$

$$f_{SBM}(X_v) = \frac{Y_{tr}^T \cdot (X_{tr}^T \otimes X_v)^{-1} \cdot (X_{tr}^T \otimes X_v)}{\mathbf{1}^T \cdot (X_{tr}^T \otimes X_v)^{-1} \cdot (X_{tr}^T \otimes X_v)} \quad \textbf{(2.49)}$$

Finalmente, se evalúa X_v como perteneciente a clase positiva si Y_v supera cierto umbral, el cual se puede ajustar para maximizar alguna métrica en particular. La implementación efectiva de SBM con kernel no lineal requiere variar los parámetros de éste, escogiendo la combinación con mejor desempeño sobre la base de validación.

2.4.4. Métricas de desempeño

Los resultados del desempeño o rendimiento de un modelo suelen ser tabulados en una matriz de confusión, también conocida como matriz de error o tabla de contingencia. Su objetivo es facilitar un análisis más detallado de los resultados. En la Ilustración 2.18 se muestra una matriz de confusión de forma genérica, donde TP y TN hacen referencia a verdaderos positivos (*True Positive*) y verdaderos negativos (*True Negative*) respectivamente, es decir, la cantidad de resultados bien clasificados con etiquetas positivas y negativas. Por otra parte, FP y FN se refieren a falsos positivos (*False Positive*) y falsos negativos (*False Negative*) respectivamente, representando la cantidad de ejemplos mal clasificados con etiquetas positivas y negativas.

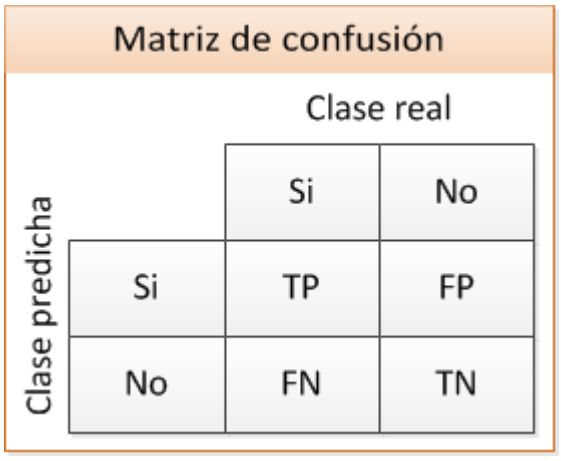

Ilustración 2.18: Matriz de confusión predictiva

La métrica más popular es *Accuracy* [41], ésta corresponde al porcentaje de predicciones bien clasificadas. Tiene las propiedades de ser una métrica simétrica, puede ser utilizada con múltiples clases y varía entre cero (clasificación errónea total) y uno (clasificación perfecta). Su popularidad radica en el bajo costo computacional en su cálculo y su fácil interpretación. A partir de la matriz de confusión predictiva, su ecuación se expresa en (2.50).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \textbf{(2.50)}$$

Otras métricas ampliamente utilizadas [42] [43] son:

- ***Precision*:** Representa que tantos valores clasificados positivos son relevantes. Una forma de malinterpretar y maximizar este valor es entregando sólo el valor en que el predictor tenga mayor confianza. Su ecuación se expresa en (2.51).

$$Precision = \frac{TP}{TP + FP} \quad \textbf{(2.51)}$$

- ***Recall*:** También llamada exhaustividad, representa qué tan bueno es un test o predictor en detectar valores positivos. Se puede malinterpretar y maximizar si el predictor retorna siempre positivo. Su ecuación se formula en (2.52).

$$Recall = \frac{TP}{TP + FN} \quad \textbf{(2.52)}$$

- ***F-Score:*** También llamada medida-F, corresponde a un balance entre *Precision* y *Recall*. El parámetro β determina qué tanta importancia se le da a cada una de las anteriores, si $\beta > 1$ se da mayor importancia a $Precision$ y viceversa. Escoger $\beta = 1$ se suele llamar media harmónica, pues se da la misma ponderación a los estadísticos que la componen. Su fórmula se presenta en (2.53).

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall} \quad \textbf{(2.53)}$$

La curva de Característica Operativa del Receptor (*Receiver Operating Characteristic* <ROC>) es otra métrica vastamente utilizada para la evaluación de rendimiento de un modelo [44]. Ésta consiste en un espacio dos-dimensional

con las razones de falsos positivos (*False Positive Rate* <FPR>) para la abscisa y verdaderos positivos (*True Positive Rate* <TPR>) para la ordenada.

Un clasificador perfecto se situaría en la esquina superior izquierda, sin falsos positivos ni falsos negativos. Por el contrario, un clasificador completamente aleatorio se situaría en un punto a lo largo de la diagonal (Ilustración 2.19). La curva ROC se puede utilizar para generar diversos estadísticos de rendimiento, siendo el más utilizado el área bajo la curva ROC, también llamado AUC (*Area Under Curve*). Éste se puede interpretar como la probabilidad de que un clasificador evalúe una clase positiva elegida aleatoriamente, mayor que una negativa.

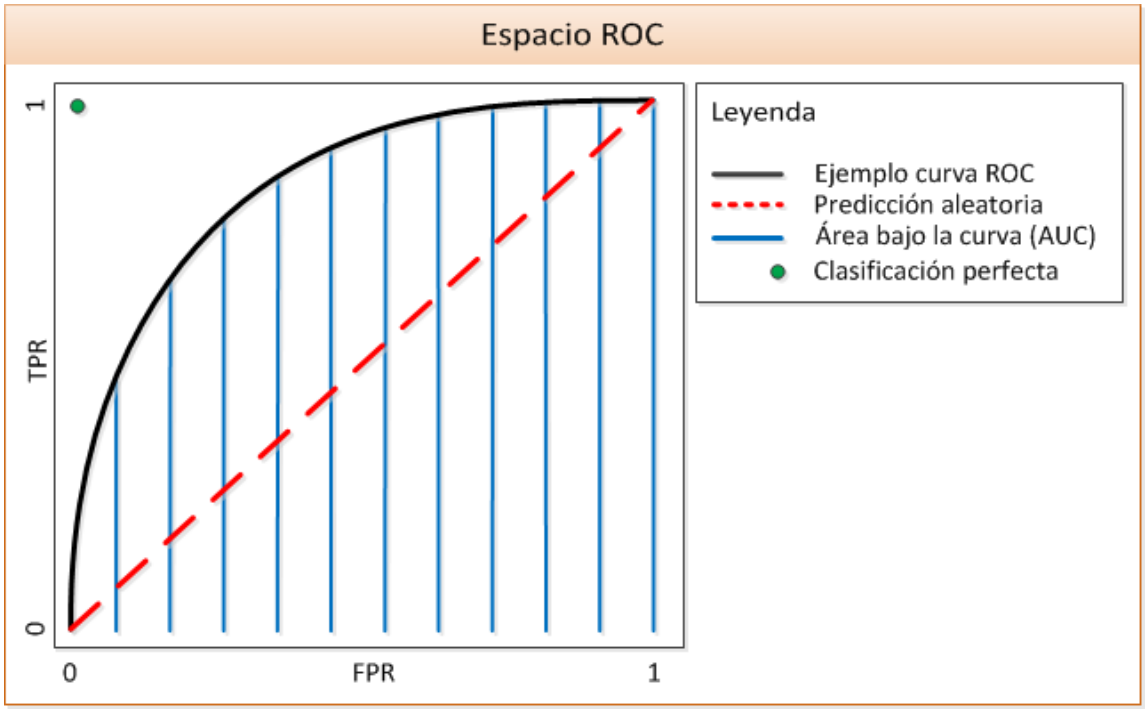

Ilustración 2.19: Espacio ROC

Capítulo 3

Generación de base de datos

En el presente capítulo se expone de forma detallada los pasos realizados para la preparación de la base de datos, en particular se describen las tres primeras etapas del proceso KDD (sección 2.3).

3.1. Selección de datos

Con el objetivo de implementar un sistema robusto y representativo, se decide utilizar los datos transaccionales de las 40 acciones que conforman el IPSA de Chile el 2013 (Anexo A). Se desea extraer variables utilizando información de las transacciones de Insiders, por lo tanto, se requiere que los datos se encuentren en períodos diarios.

La fuente utilizada para extraer la información OHLC de las acciones es la Corredora de Bolsa de Consorcio [45]. Ésta posee los datos desde la creación de las empresas que componen el índice con períodos diarios. Por otra parte, se utiliza la base de datos de Insiders Chile [46] con las transacciones de Insiders, la cual posee datos de operaciones entre 04-01-2010 y 06-05-2013.

3.2. Pre-procesamiento: Limpieza y estructuración

En esta fase se corrigen inconsistencias en los datos y se dejan con la estructura necesaria para trabajar con éstos. En la base de datos de acciones se encuentran valores nulos o faltantes para el precio de cierre en períodos anteriores al año 2010. Se decide reemplazar éstos por el precio de apertura del período siguiente. En fechas anteriores al 2005, se encuentra que los precios *High* y *Low* no siempre corresponden al máximo y mínimo respectivamente, por lo que son reemplazados con el precio correspondiente. Para cada activo se genera una estructura con los siguientes campos:

- Empresa:
    - Precios
        - *Open*: Precio de apertura.
        - *High*: Precio máximo.
        - *Low*: Precio mínimo.
        - *Close*: Precio de cierre.

    - Información:
        - *Ticker*: Nemotécnico bursátil que representa una acción.
        - Nombre: Nombre de la razón social.

En la base de datos de Insiders se encuentra que los *ticker* no corresponden a los utilizados por las empresas, además de desórdenes en los datos. Los *ticker* son cambiados por los que corresponden y se dejan los datos ordenados según empresa. Los campos considerados de importancia por transacción son los siguientes:

- *Ticker*: Nemotécnico bursátil que representa la acción tranzada.
- Fecha *In*: Fecha en la que el Insider le dio aviso al SVS de su transacción.
- Tipo de Transacción:
    - Adquisición: Compra de la acción.
    - Enajenación: Venta de la acción.
    - SVPE: Suscripción de valores primera emisión (compra).
- Cantidad: Número de acciones transadas por el Insider.
- *Price*: Precio de cierre del día de la acción transada.
- *Amount*: Monto total transado considerando precio de cierre.

Se decide utilizar el precio de cierre diario en vez del precio de compra original, ya que generalmente pasan de días hasta años antes que un Insider enajene las acciones previamente adquiridas, por lo que no se tratan de operaciones *intraday* (alta frecuencia).

3.3. Análisis de ZigZag

Se realiza un análisis del comportamiento del indicador ZigZag para distintos parámetros de retracciones porcentuales mínimas RP. Con el fin de obtener un resultado representativo, las pruebas se realizan para las 40 acciones del IPSA durante el período de entrenamiento (Tabla 3.2). Se varía el parámetro de retracción mínima RP entre $RP_{ini} = 0.5\%$ y $RP_{fin} = 15\%$ utilizando intervalos de $\nabla = 0.5\%$, además, se consideran por separado los ZigZag de alza, baja y ambos juntos. Para facilitar la visualización y análisis de los datos, se muestran los estadísticos de promedio y la mediana para cada valor, así se comprende de manera cualitativa la distribución de los resultados sin realizar un gráfico tridimensional.

En la Ilustración 3.1 se grafica los períodos de diferencia entre indicaciones de ZigZag para distintos parámetros de RP de ZigZag, en ésta se observa un comportamiento similar en promedio y mediana tanto para alza como para baja, apreciando una variación casi lineal de los estadísticos en todo

el dominio. Por otra parte, al ser la mediana menor que el promedio, se infiere que la mayoría de los ZigZag tienen una duración menor al promedio, siendo el promedio afectado directamente por ZigZag con períodos fuera de rango, es decir, de muy larga duración. Finalmente, se distingue que para el rango de RP comprendido entre 5% y 10%, en promedio se debe esperar más períodos para que suceda una retracción de alza que para una retracción baja.

En la Ilustración 3.2 se expone el cambio porcentual de las acciones para distintos valores de RP de ZigZag. Por construcción se obtiene que el cambio porcentual de precio siempre es mayor o igual que el parámetro de ZigZag. Se aprecia que el cambio porcentual del precio varía de forma casi lineal con una pendiente de 2.5654 en promedio y de 2.0088 en mediana (ver Tabla 3.1), por lo que, si se espera una retracción del precio de cierto porcentaje, es recomendable esperar que el precio haya tenido un cambio porcentual de al menos las veces señaladas por la mediana en comparación al último ZigZag, esto es equivalente a decir que es más fácil seguir con la tendencia que cambiarla. Por otra parte, se distingue que los movimientos de alza tienden a tener un cambio porcentual levemente mayor que los de baja, además, al estar la mediana por debajo del promedio, se infiere que la mayoría de las retracciones tienden a tener cambios porcentuales menores que el promedio, el que se ve afectado por retracciones porcentuales fuera de rango.

Finalmente, se distingue que los cambios porcentuales de alza tienden a ser mayores en promedio con $RP \leq 8\%$, pero con $RP \geq 8.5\%$ los cambios de baja tienden a ser mayores. De lo que se infiere que, a pequeñas retracciones, se tiende ir más hacia el alza antes de cambiar tendencia; por el contrario, a grandes retracciones, los movimientos de baja tienden a tener un mayor cambio porcentual antes que surja una nueva alza. Es decir, las grandes caídas de las acciones son mayores que las grandes alzas en retorno promedio.

En la Ilustración 3.3 se grafican los cambios porcentuales para distintos períodos en promedio y mediana (combinación de recorridos de Ilustración 3.1 e Ilustración 3.2). Se aprecia que éstos poseen un comportamiento casi lineal para períodos de diferencia sobre 20, umbral en el cual se comienza a observar una tendencia de cambios porcentuales mayores hacia la baja. Cabe destacar que, a causa del comportamiento casi lineal observado y que los retornos acumulados varían de forma compuesta (no simple), es más conveniente extraer retornos pequeños a alta frecuencia que retornos grandes a baja frecuencia.

En la Tabla 3.1 se detallan los parámetros de una regresión lineal de diferencia de períodos y cambio porcentual para distintos parámetros de retracción porcentual de ZigZag. Ésta se realiza utilizando alza y baja a la vez (equivalente a “general” en Ilustración 3.1 e Ilustración 3.2). Se considera una buena aproximación la utilización de los datos generales y de un modelo de

aproximación lineal debido al comportamiento previamente descrito de dichas métricas.

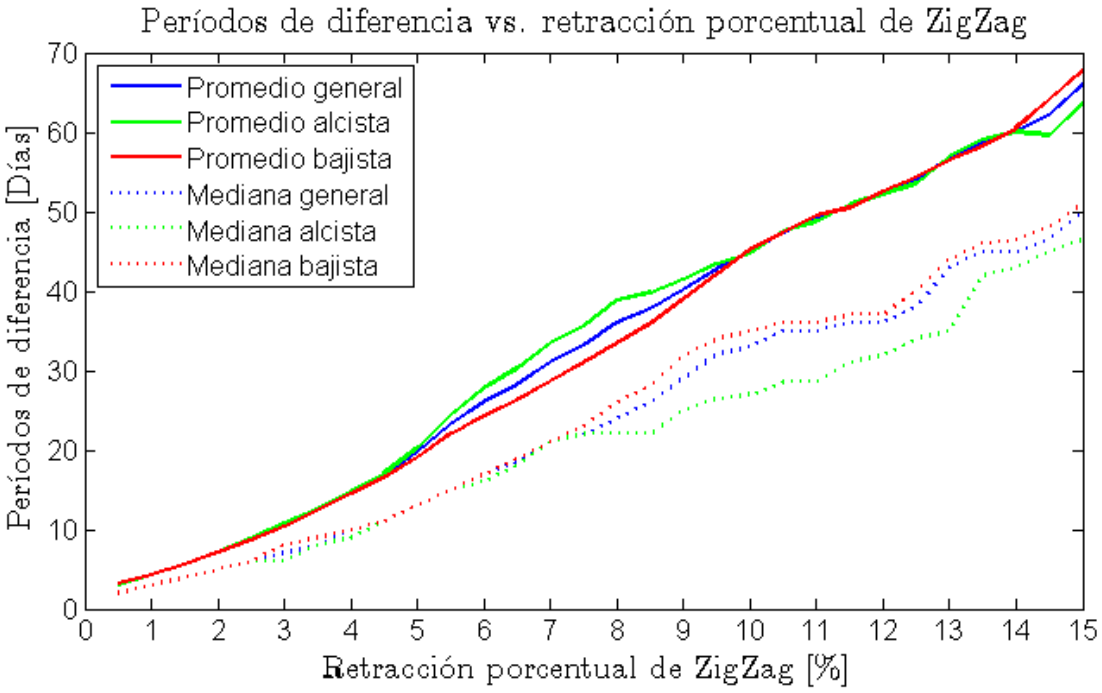

Ilustración 3.1: Diferencia de períodos para distintos parámetros de retracción de ZigZag

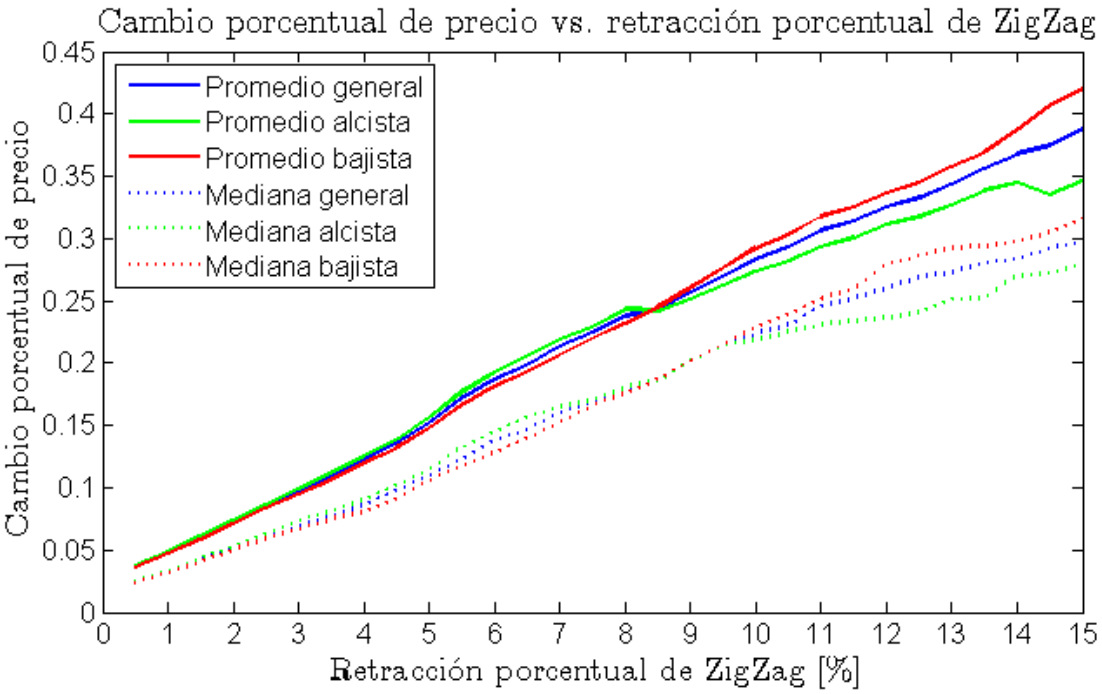

Ilustración 3.2: Cambio Porcentual para distintos parámetros de retracción de ZigZag

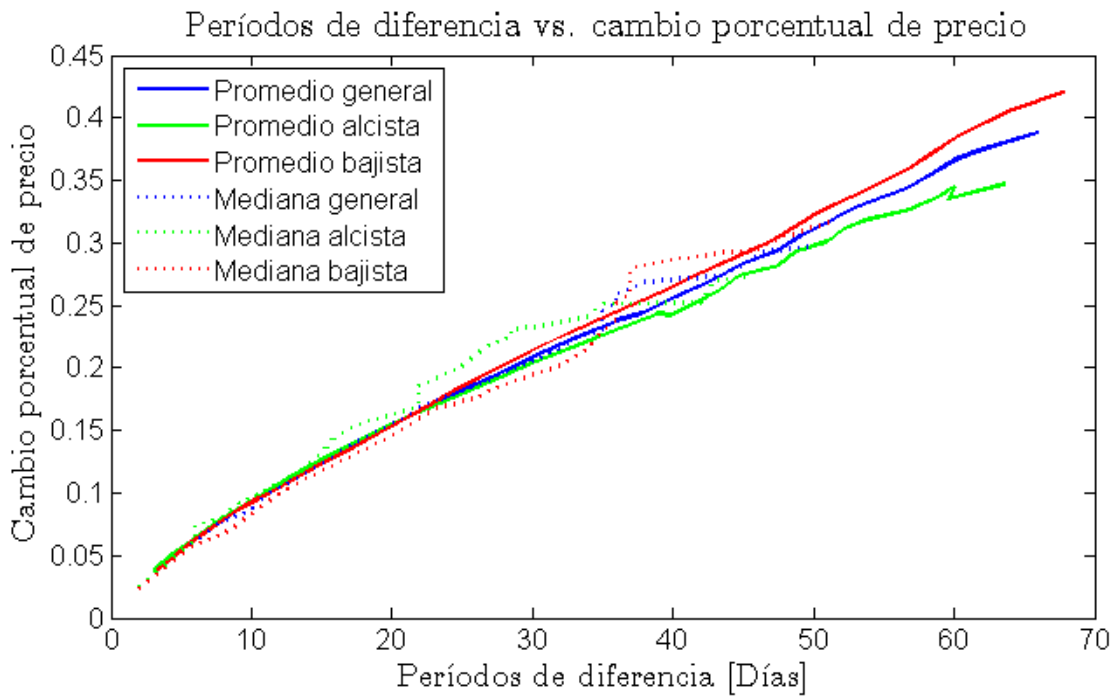

Ilustración 3.3: Cambio Porcentual para distintos períodos de diferencia en promedio y mediana

Regresión Lineal	Diferencia de Períodos		Cambio Porcentual	
	Promedio	Mediana	Promedio	Mediana
Pendiente	449.5644	339.6663	2.4654	2.0088
Sesgo	-1.3759	-2.4575	0.0292	0.0136

Tabla 3.1: Parámetros de regresión lineal para análisis de ZigZag (general)

3.4. Transformación

3.4.1. Variables independientes

Las variables independientes corresponden a atributos extraídos de los datos previamente procesados. Su fin es obtener patrones a partir de éstos en la fase de minería de datos, bajo la premisa que poseen estructura predictiva para anticipar cambios de tendencia del mercado. En particular se extraen doce variables de Insider, 72 de MACD, 40 de RSI y 150 de SO, sumando un total de 274 atributos.

3.4.1.1. Variables extraídas a partir de Insiders

Se comprimen los datos de Insider (informados en la SVS) para cada acción, de manera que sólo se considera la información macro de éstos. En particular, para cada período $\boldsymbol{t}$ se tiene (3.1), donde $\boldsymbol{k} \in \{Long, Short\}$ en función de (3.2), es decir, se consideran las operaciones de adquisición y enajenación por separado, sumando para cada período montos de tranzados por los agentes (para cada activo por separado).

$$VolumeInsider_t^{(k)} = \sum_{i==t} Amount_i^{(k)} \quad \textbf{(3.1)}$$

$$tipoOperación_i = \begin{cases} Long & TipoTransacción_t \in \{Adquisición, SVPE\} \\ Short & TipoTransacción_t \in \{Enajenación\} \end{cases} \quad \textbf{(3.2)}$$

De los datos de Insider se extraen doce atributos a ser utilizados en la fase de minería de datos, siendo la mitad representativas de movimiento de alza o compra y la otra mitad de baja o venta, estas son:

- **Al menos una operación**

Se genera una variable binaria que representa si para cada período de embebido $\boldsymbol{t}$ existe o no una operación $\boldsymbol{k}$ (diferenciada para *Long* y *Short*).

$$AtLeastOne_t^{(k)} = \begin{cases} 1 & VolumeInsider_t^{(k)} > 0 \\ 0 & \sim \end{cases} \quad \textbf{(3.3)}$$

- **Cantidad vecindad**

Representa si en el último tiempo ha habido transacciones por parte de Insiders. El objetivo de ésta es interiorizar el hecho de que éstos tranzan esperando retornos positivos en el futuro y no para sólo un período (no realizan operaciones *Intraday*). Corresponde al indicador $AtLeastOne$ desplazado en el tiempo considerando días transados. Se diferencia por tipo de operación $\boldsymbol{k}$ y utiliza el parámetro $vecindad$, el que representa la cantidad de períodos a considerar en el pasado. Su ecuación se expresa en (3.4) y su valor final normalizado en (3.5). En particular se utiliza $vecindad = 5$, es decir, se consideran las transacciones de los últimos cinco días hábiles.

$$NeighborhoodQuantity_t^{(k)}(vecindad) = \sum_{t-vecindad}^{t} AtLeastOne_t^{(k)} \quad \textbf{(3.4)}$$

$$Neighborhood_t^{(k)}(vecindad) = \frac{NeighborhoodQuantity_t^{(k)}(vecindad)}{\max_t NeighborhoodQuantity_t^{(k)}(vecindad)} \quad \textbf{(3.5)}$$

- **Operaciones con mayor vecindad**

Representa los períodos con mayor vecindad, es decir, si ha habido muchos períodos en que los Insiders han realizado el mismo tipo de operación en la cercanía temporal. Es equivalente a $Neighborhood$ considerando sólo los períodos de mayor actividad y el parámetro $clustering$ representa el porcentaje de períodos tranzados con más cercanía temporal. En la ecuación (3.6), $v_t^{(k)}$ representa si hubo alguna transacción en el período $\boldsymbol{t}$, además se consideran sólo el porcentaje de $clustering$ valores aproximado hacia arriba (3.7). Finalmente, en la fórmula (3.8) se consideran sólo los períodos con mayor valor.

$$v_t^{(k)} = \begin{cases} 1 & Neighborhood_t^{(k)}(vecindad) > 0 \\ 0 & \sim \end{cases} \quad \textbf{(3.6)}$$

$$\#valores^{(k)}(clustering) = \left\lceil clustering \cdot \sum_{t=1} v_t^{(k)} \right\rceil \quad \textbf{(3.7)}$$

$$Clusterizest_t^{(k)} = \bigcup_{j=1}^{\#valores^{(k)}(clustering)} \max_t Neighborhood_t^{(k)}(vecindad) \quad \textbf{(3.8)}$$

- **Logaritmo de volumen Insider**

Dado que el monto total tranzado por los agente en un día puede variar bastante, se genera una variable que considera el volumen transado de forma logarítmica (3.9). Se suma la unidad para eliminar valores negativos y se normaliza como se expresa en (3.10).

$$logAmountAux_t^{(k)} = \log\left(1 + VolumeInsider_t^{(k)}\right) \quad \textbf{(3.9)}$$

$$logAmount_t^{(k)} = \begin{cases} \dfrac{logAmountAux_t^{(k)}}{\max_t logAmountAux_t^{(k)}} & \max_t logAmountAux_t^{(k)} > 0 \\ 0 & \sim \end{cases} \quad \textbf{(3.10)}$$

- **Monto sobre el promedio**

Esta variable representa si para un período en particular $\boldsymbol{t}$ se ha transado mayor volumen (3.12) que el volumen promedio de los períodos transados (3.11).

$$AmountMean^{(k)} = \frac{1}{\sum_t AtLeastOne_t^{(k)}} \sum_{t=AtLeastOne_t^{(k)}>0} VolumeInsider_t^{(k)} \quad \textbf{(3.11)}$$

$$aboveMean_t^{(k)} = \begin{cases} 1 & VolumeInsider_t^{(k)} > AmountMean^{(k)} \\ 0 & \sim \end{cases} \quad \textbf{(3.12)}$$

- **Monto sobre la mediana**

Esta variable es equivalente a $aboveMean$, sin embargo debido a que el volumen transado puede variar mucho, se utiliza la mediana como estadístico de umbral (3.13), siendo así más representativo de los datos. Al igual que en el caso anterior sólo se consideran los períodos tranzados.

$$aboveMedian_t^{(k)} = \begin{cases} 1 & VolumeInsider_t^{(k)} > median(VolumeInsider_t^{(k)} > 0) \\ 0 & \sim \end{cases} \quad \textbf{(3.13)}$$

3.4.1.2. Atributos extraídos a partir de indicadores

Se extraen variables o atributos de distintos indicadores, en particular del MACD, RSI y SO con distintos conjuntos de parámetros. La elección de estos indicadores se respalda en su capacidad para representar cuantitativamente el momento, dirección y tendencia de los activos.

3.4.1.2.1. Variables MACD

Se extraen variables de dos conjuntos de parámetros de MACD, en particular se utiliza $(12{,}26{,}9)$ y $(5{,}35{,}5)$ para los valores de $(EMA\ rápida,\ EMA\ lenta,\ MACD\ EMA)$. Con estos conjuntos de valores se calcula la línea, señal e histograma MACD con la ecuaciones expresadas de (2.7) a (2.9).

Para cada conjunto de parámetros se extraen 36 variables, siendo cuatro variables de cruces y 32 de divergencia para MACD e histograma, obteniendo 18 señales alcistas y 18 bajistas. En el MACD las señales de alza y baja son simétricas, por lo que se exponen las alcistas a continuación:

- Cruces:
  - $MACDcrossUp$: Línea de MACD cruza por sobre cero.
  - $signalCrossUp$: Señal cruza por sobre línea de MACD (equivalente a que Histograma cruce por sobre nivel cero) mientras línea MACD es negativa.
- Divergencias:
  - Divergencias del tipo alcista con doble indicador, siendo la línea MACD el principal y la de señal el secundario.
  - Divergencias del tipo *alcista* con el histograma.

Para ambas divergencias los parámetros utilizados son:

- $lastRegularPeaks = Inf$: Se consideran todos los *peaks*.
- $lastHiddenPeaks = Inf$: Se consideran todos los *peaks*.
- $lastPeriods = Inf$: Se consideran todos los períodos.
- $searchNeighborhood = 2$: Se consideran dos períodos por lado.
- $maximumPeriodDifference = 4$: Relación uno a cuatro para div. triple.
- $overboughtThreshold = -Inf$: No posee umbral mínimo.
- $offset = 0$: No hay desfase.

3.4.1.2.2. Variables RSI

Se extraen variables del indicador RSI, para esto, se utilizan las ecuaciones (2.10) a (2.13) con $n = 14$. Se utilizan los umbrales de sobrecompra de 70 y 80 (umbrales de sobreventa de 30 y 20 respectivamente). Se obtienen cuatro variables de cruce, cuatro de agotamiento y 32 de divergencia, sumando un total de 40 variables.

- Cruces:
  - $CrossUp20$: RSI cruza por sobre 20 (Señal alcista).
  - $CrossUp30$: RSI cruza por sobre 30 (Señal alcista).
  - $CrossDown80$: RSI cruza por bajo 80 (Señal bajista).
  - $CrossDown70$: RSI cruza por bajo 70 (Señal bajista).
- Agotamiento:
  - $extUnder20$: RSI está bajo 20 (Señal de sobreventa).
  - $extUnder30$: RSI está bajo 30 (Señal de sobreventa).
  - $extAbove80$: RSI está sobre 80 (Señal de sobrecompra).
  - $extAbove70$: RSI está sobre 70 (Señal de sobrecompra).
- Divergencias:
  - Dos divergencias RSI con los siguientes parámetros:
    - $lastRegularPeaks = Inf$.
    - $lastHiddenPeaks = Inf$.

- $lastPeriods = Inf$
- $searchNeighborhood = 2$
- $maximumPeriodDifference = 4$
- $overboughtThreshold = 70\ y\ 80$: Se evalúan ambos.
- $offset = 100$: Desfase de 100, es decir umbrales 30 y 20 para divergencias alcistas.

3.4.1.2.3. Variables SO

Se extraen variables de tres conjuntos de parámetros de SO, en particular, con los valores $(5,3,3)$, $(14,3,3)$ y $(21,3,3)$ para $(\%K\ period, \%D\ period, Slowing)$. Su objetivo es extraer los cambios de tendencia a corto, mediano y largo plazo. Se utilizan los umbrales de 70 y 80 para sobrecompra, por simetría se utiliza 30 y 20 para sobreventa. Para cada parametrización se extraen 50 variables, resultado en un total de 150 variables de SO.

- Cruces:
  - $CrossUp20D$: $\%D$ cruza por sobre 20 (Señal alcista).
  - $CrossUp20K$: $\%K$ cruza por sobre 20 (Señal alcista).
  - $CrossUp30D$: $\%D$ cruza por sobre 30 (Señal alcista).
  - $CrossUp30K$: $\%K$ cruza por sobre 30 (Señal alcista).
  - $CrossDown80D$: $\%D$ cruza por bajo 80 (Señal bajista).
  - $CrossDown80K$: $\%K$ cruza por bajo 80 (Señal bajista).
  - $CrossDown70D$: $\%D$ cruza por bajo 70 (Señal bajista).
  - $CrossDown70K$: $\%K$ cruza por bajo 70 (Señal bajista).
  - $CrossSignalUp$: $\%K$ cruza por sobre $\%D$ mientras $\%D < 50$.
  - $CrossSignalDown$: $\%K$ cruza por bajo $\%D$ mientras $\%D > 50$.
- Agotamiento:
  - $extUnder20D$: $\%D$ está bajo 20 (Señal de sobreventa).
  - $extUnder20K$: $\%K$ está bajo 20 (Señal de sobreventa).
  - $extUnder30D$: $\%D$ está bajo 30 (Señal de sobreventa).
  - $extUnder30K$: $\%K$ está bajo 30 (Señal de sobreventa).
  - $extAbove80D$: $\%D$ está sobre 80 (Señal de sobrecompra).
  - $extAbove80K$: $\%K$ está sobre 80 (Señal de sobrecompra).
  - $extAbove70D$: $\%D$ está sobre 70 (Señal de sobrecompra).
  - $extAbove70K$: $\%K$ está sobre 70 (Señal de sobrecompra).
- Divergencia:
  - Dos divergencias de SO con los siguientes parámetros:
    - $lastRegularPeaks = Inf.$
    - $lastHiddenPeaks = Inf.$
    - $lastPeriods = Inf.$
    - $searchNeighborhood = 2.$
    - $maximumPeriodDifference = 4.$

- $overboughtThreshold = 70\ y\ 80$: Se evalúan ambos.
- $offset = 100$: Desfase de 100, es decir, umbrales 30 y 20 para divergencias alcistas.

3.4.1.2.4. Variables de divergencia

Las variables de divergencia se pueden calcular para uno o dos indicadores, en el caso de dos indicadores, el indicador secundario corresponde al filtrado o MA del indicador principal. Se pueden extraer las variantes de divergencia mostradas Ilustración 2.8 y la versión triple apreciada en la Ilustración 2.9. Se incluye una variante de divergencia con confirmación simple y doble, para uno y dos períodos con movimiento del indicador a favor de la señal divergente respectivamente, teniéndose así las siguientes variantes:

- Divergencia alcista o bajista.
- Patrón divergente regular u oculto.
- Continuación normal o triple.
- Confirmación simple o doble.

Si se consideran las variantes expuestas se tienen $16\ (2^4)$ divergencias por indicador, siendo ocho indicadores de alza y ocho de baja.

En Anexo C se expone el pseudocódigo para calcular todos los tipos de divergencias. En la Ilustración se presentan los parámetros, entradas, salidas e inicialización general del algoritmo. En la Ilustración e Ilustración se expresa el pseudocódigo para la obtención de patrones divergentes regulares y ocultos respectivamente, además, se desarrollan los códigos auxiliares en la Ilustración C.. Finalmente, en la Ilustración se detalla la forma de obtener divergencias de continuación triple utilizando como argumento divergencias de continuación simple.

Cabe destacar que las variables de divergencia se aplican al precio de cierre del valor del indicador, por lo que la detección de patrones divergentes se obtiene en este período. Además, se consideran los desfases de uno o dos períodos para las divergencias con confirmación “simple” y “doble” respectivamente

Con el fin que la detección de divergencia sea más robusta y menos paramétrica, para cada período no se considera el precio máximo local, sino que se busca el precio máximo en un vecindario (parametrizado).

En la Ilustración 3.4 se presenta un ejemplo de obtención de divergencia regular bajista a partir del pseudocódigo expuesto en Anexo C, en particular, hace referencia al proceso expuesto en la Ilustración . Se considera

el momento actual remarcado por la línea vertical color naranjo, siendo necesario repetir el proceso para cada período. Los pasos seguidos en el ejemplo para la obtención de divergencias bajistas regulares son:

i. Se busca el precio máximo P_0 en torno a la vecindad temporal del momento actual. Por causalidad, sólo se consideran períodos anteriores.
ii. Se recorren los precios hacia el pasado hasta encontrar un precio mayor al actual, el que se encuentra en P_4, utilizando este período como límite de búsqueda de divergencias regulares (línea vertical morada). Este límite se impone para optimizar recursos, ya que, por construcción, las divergencias regulares requieren que el precio pasado sea menor que el precio actual.
iii. Se recorre el indicador hacia el pasado evaluando la existencia de *peaks*, los que se definen como períodos con valor mayor que dos períodos hacia el pasado y dos períodos hacia el futuro. Se comienza en el momento actual y se termina en el límite de búsqueda. Se encuentran los *peaks* I_0, I_1, I_2, I_3 e I_4, siendo I_1 omitido ya que es menor que el *peak* I_0 encontrado previamente.
iv. Se busca el precio máximo en un vecindario temporal para los *peaks* válidos, obteniéndose P_0, P_2, P_3y P_4.
v. Se evalúa la existencia de divergencias respecto a I_0 y su precio P_0, la condición de divergencia regular requiere un valor de indicador y de precio que cumpla $(P_X < P_0)\,\&\,(I_X > I_0)$. Por construcción, todos los valores de indicador cumplen la desigualdad, por lo que se deben evaluar los precios. Se aprecia que $P_4 > P_0$, por lo que es descartado; quedando sólo P_2 y P_3 válidos como divergencias.
vi. Se analiza si existe alguna intersección entre las rectas de precio y de indicador, con sus respectivas series de tiempo. Se aprecia que la recta $\overline{P_0P_2}$ intersecta la serie de tiempo de precios, por lo que la divergencia es descartada. Encontrándose así sólo la divergencia regular bajista entre I_0 e I_3.

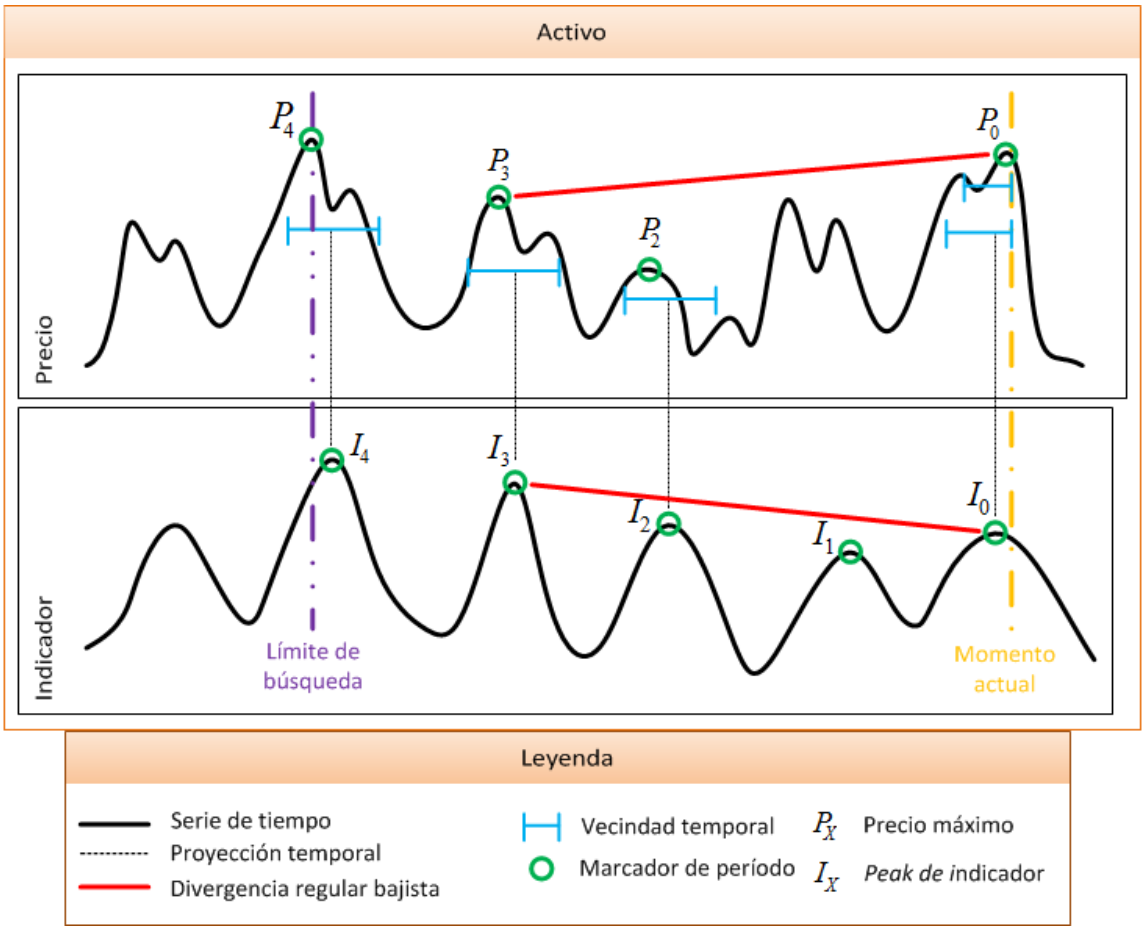

Ilustración 3.4: Ejemplo obtención de divergencia regular bajista

La búsqueda de divergencias ocultas bajistas se realiza con la misma metodología, pero obteniendo el límite de exploración en la serie de indicador (en vez de la serie de precios), descartando períodos con precios menores al actual (en vez de *peaks* de indicador) y cambiando la condición de divergencia por $(P_X > P_0)$ & $(I_X < I_0)$. Por otra parte, la obtención de divergencias alcistas es equivalente a la metodología propuesta, pero con ambas series de tiempo invertidas, es decir, $Indicador = -Indicador$ y $Precio = -Precio$.

3.4.2. Variable dependiente

3.4.2.1. Generación de variable dependiente a partir de ZigZag

La variable dependiente corresponde a la variable objetivo que se desea los atributos o variables independientes puedan predecir o ajustarse en la fase de minería de datos. Dado que se desea predecir cambios de tendencia o equivalente, recomendaciones de compraventa, a partir de máximos locales (períodos de redistribución) o mínimos locales (períodos de acumulación) de la serie de tiempo del activo en estudio, se propone la utilización del indicador de ZigZag aplicado al precio de cierre. Por definición éste corresponde a cambios de tendencia o retracciones de al menos cierto umbral.

La propuesta de utilización de ZigZag se realiza bajo la hipótesis de que el mercado no posee comportamiento oscilatorio con una frecuencia fija, por lo que fijar un período estático (por ejemplo 30 o 90 días), es sobre-ajustarse a los datos y no extraer todo el retorno posible. Por otra parte, la utilización de cambios porcentuales estáticos tiene la ventaja de que no se exige períodos fijos y la desventaja de que pueden suceder varias veces en la misma dirección. Por lo tanto, utilizar el ZigZag, que exige un cambio porcentual mínimo, sin sobre-ajustarse a ventanas de tiempo ni cambios porcentuales estáticos, se considera una buena elección.

El indicador ZigZag puede ser aplicado al precio de cierre o al máximo-mínimo de un período, siendo ésta última descartada ya que, si se utiliza parámetro de retracción pequeño, en muchos casos se pueden tener dos indicaciones de ZigZag en un mismo período. Efecto no deseado, ya que no se trabaja con datos *intraday*.

El pseudocódigo implementado de ZigZag aplicado al precio de cierre se muestra de forma esquemática en el Anexo B. Se calcula el indicador ZigZag aplicado al precio de cierre dado un parámetro de retracción porcentual mínima RP, luego se generan las recomendaciones de compra y venta de acuerdo a las ecuaciones (3.14) y (3.15) respectivamente, los cuales sólo marcan el período de cambio de ZigZag.

$$LongZZ_t = \begin{cases} 1 & Close(ZZ_t) < Close(ZZ_{t-1}) \\ 0 & \sim \end{cases} \quad \textbf{(3.14)}$$

$$ShortZZ_t = \begin{cases} 1 & Close(ZZ_t) > Close(ZZ_{t-1}) \\ 0 & \sim \end{cases} \quad \textbf{(3.15)}$$

Con el fin de comprender de mejor manera cómo se comportan las variables dependientes extraídas de ZigZag, se estudia la distribución de etiquetas positivas para ambas bases (aplicado a las 40 acciones en estudio) con distintos valores de retracción porcentual. Los resultados del análisis se muestran en Ilustración 3.5, como era de esperarse, se tiene prácticamente la misma cantidad de etiquetas para ambas clases en todo el dominio, siendo la diferencia nunca mayor a uno e imperceptible porcentualmente. Esto se atribuye a que siempre después de un ZigZag de alza viene uno de baja y viceversa, siendo una propiedad intrínseca del indicador. Por otra parte, se observa que a medida que aumenta el parámetro de retracción del ZigZag, disminuye la proporción de etiquetas positivas, esto se debe a que las acciones poseen mayor cantidad de retracciones pequeñas. Finalmente, se aprecia que la cantidad de etiquetados positivos es muy pequeña, convirtiéndose en un problema de minería de datos con clases muy desbalanceadas, requiriendo muchos datos para poder ser aplicado correctamente.

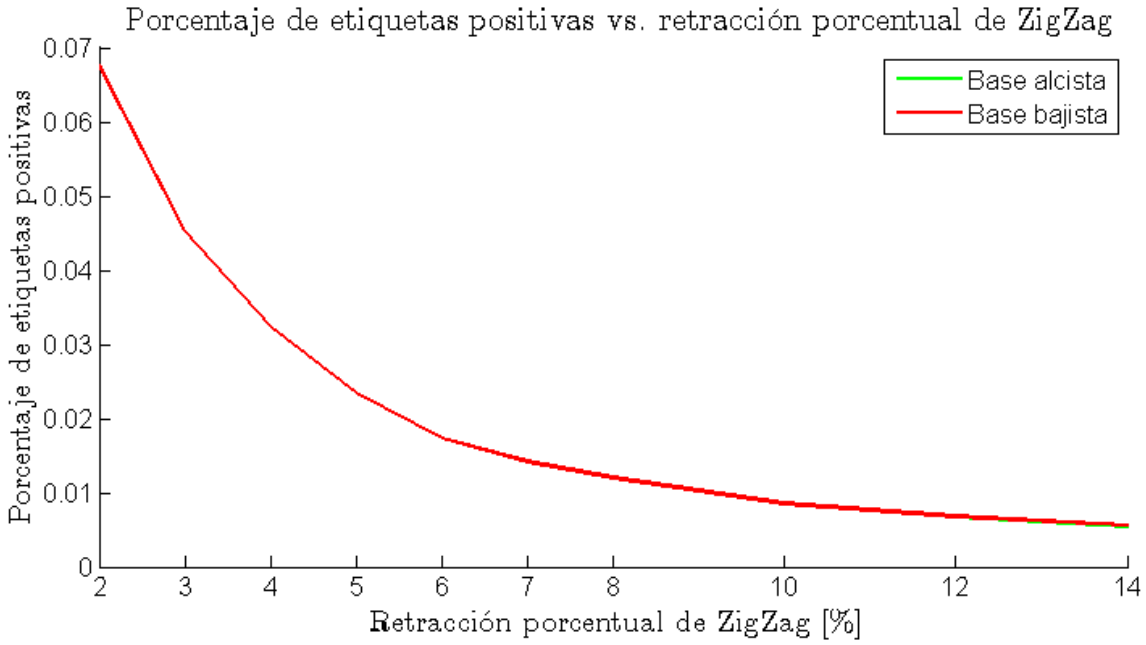

Ilustración 3.5: Distribución de etiquetas positivas a partir de ZigZag

3.4.2.2. Extensión de variable dependiente a partir de ZigZag

Se desea que más períodos sean etiquetados como adecuados para operaciones compraventa, con el objetivo de poder aplicar apropiadamente herramientas de aprendizaje de máquina en la fase de entrenamiento. Para esto, se debe desarrollar un método que mantenga las propiedades deseadas de etiquetado, es decir, que indique mínimos locales para compra y máximos locales para venta, con el objetivo de obtener el mayor retorno posible sin aumentar el riesgo en las operaciones.

Se desarrolla un algoritmo a partir de las variables dependientes ya generadas utilizando ZigZag (Anexo C), éste realiza extensiones no causales a partir de las etiquetas ya generadas, recomendando entrar o salir de una operación antes que el precio varíe en gran proporción.

Como se ejemplifica en la Ilustración 2.7, el indicador ZigZag señala como apropiado sólo un período de compra o venta por cada retracción, si éstos son considerados como un buen punto de partida, se puede analizar la vecindad temporal y así evaluar si se encuentran más recomendaciones en torno a un rango apropiado de precio.

El rango de períodos se evalúa en la dimensión temporal como un máximo porcentaje entre los dos ZigZag más próximos (hacia el futuro y el pasado). Por otra parte, el rango de precio se determina como un porcentaje de cierre máximo antes del siguiente ZigZag, resaltando que se desean retornos al futuro, por lo que el etiquetado debe ser causal.

Como se deja expuesto en 3.3, existen retracciones de ZigZag anómalas con grandes variaciones porcentuales y que pueden poseer muchos períodos antes de la siguiente retracción. Con el fin de evitar extensiones de muchos períodos o de gran variación porcentual del precio (aumenta el riesgo por no entrar al principio del movimiento de precio), se implementa la heurística que si el valor supera el promedio, sea reemplazado por la mediana.

Se generan parámetros para evaluar una variación del algoritmo, la cual consiste en acotar el rango de extensión temporal, para esto, se consideran *blobs* para cada periodo etiquetado positivo en la dimensión temporal, es decir, dada una recomendación inicial de ZigZag, se deja de extender hacia un sentido (futuro o pasado) cuando hay un período que no se encuentre en el rango de precio.

Para cumplir los requisitos expuestos, el algoritmo desarrollado involucra los siguientes parámetros:

- $\boldsymbol{maximumPeriodPercentageExtension}$**:** Extensión temporal máxima (en porcentaje de períodos) entre el ZigZag anterior y siguiente. Varía entre cero y uno.
- $\boldsymbol{maximumPricePercentageExtension}$**:** Extensión porcentual máxima de precio entre el ZigZag anterior y posterior. Varía entre cero y uno.
- $\boldsymbol{maximumPeriodMean}$**:** Decisión binaria. Si el cambio porcentual de períodos de diferencia es mayor que el promedio de éstos, se reemplaza por el promedio.
- $\boldsymbol{earlyStop}$**:** Decisión binaria. Al recorrer el ZigZag desde un punto hacia el futuro y hacia el pasado, si se detecta un período que no se encuentra dentro del rango porcentual, se deja de recorrer en ese sentido.

En la Inicialización del algoritmo de extensión de variable dependiente se calculan las diferencias temporales y cambios porcentuales promedios y medianos tanto para compra como para venta. Éstos son equivalentes a los mostrados en la Ilustración 3.1 e Ilustración 3.2 evaluados para cada acción por separado. Se decide utilizar valor general para compra y venta a causa de la linealidad y cercanía de los valores discutida en la sección 3.3.

Tras evaluar distintas configuraciones de parámetros se decide fijar $earlyStop = false$. Esta decisión se basa en que pueden haber períodos no tan cercanos al ZigZag en que el precio sea parecido al valor de principal, siendo así estos un buen punto de operación. La Ilustración 3.6 ejemplifica lo señalado, en ésta se aprecia que si el parámetro en discusión estuviera activo, las etiquetas de venta (velas japonesas rojas en el rango septiembre a noviembre del 2009) marcarían como apropiado sólo los períodos más cercanos al ZigZag, lo cual no es recomendable, ya que claramente se observan más períodos apropiados antes que el primer período señalado como no apropiado (finaliza blob). Respaldado en el mismo concepto se decide fijar $maximumPeriodPercentageExtension = 1$ y $maximumPeriodMean = false$.

Por otra parte, el parámetro $maximumPricePercentageExtension$ debe ser balanceado, un valor muy grande indica recomendaciones de compraventa cuando el precio ya ha variado mucho (aumentando el riesgo), por el contrario, un valor muy bajo es equivalente a entregar sólo el período del ZigZag (disminuye cantidad de etiquetas positivas). Se encuentra adecuado utilizar valores en el rango comprendido entre 0.15 y 0.2, siendo este último el escogido. Un valor de 0.2 es equivalente extraer entre compra y venta al menos un 60% del cambio porcentual de precio.

Finalmente, se considera que si un cambio porcentual hacia el pasado supera el de futuro, éste se acota por el de futuro, ya que el interés es extraer retornos a futuro (y no hacia el pasado).

En la Ilustración 3.6 e Ilustración 3.7 se muestra el comportamiento de extensión temporal para el etiquetado de compraventa (velas japonesas rojas para venta y verdes para compra). En la Ilustración 3.7 se ejemplifica lo señalado respecto a retracciones de ZigZag fuera de rango. En ésta, si no se reemplazara la extensión temporal por la mediana, las recomendaciones de compra a principio de Julio durarían casi un mes. Por otra parte, si no se acotara la extensión porcentual del pasado como menor o igual a la de futuro, se recomendaría vender entre mediados de Septiembre y Octubre, lo que claramente no es apropiado, ya que el ZigZag siguiente posee un pequeño cambio porcentual.

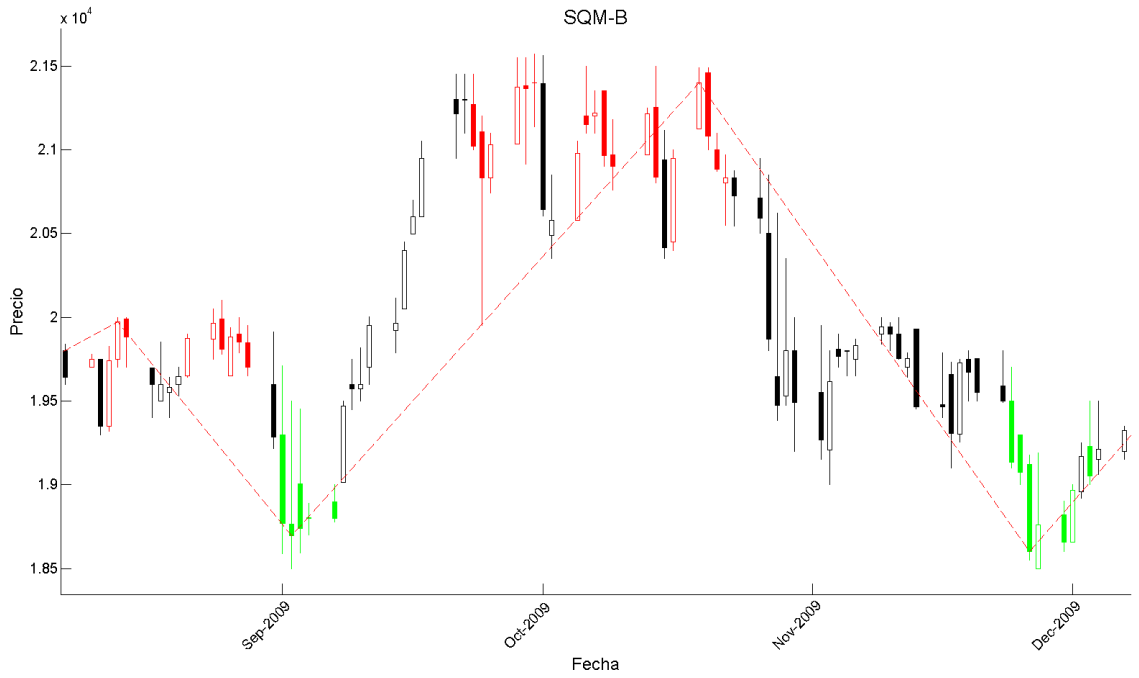

Ilustración 3.6: Extensión temporal de compraventa mostrando buen comportamiento de reventa

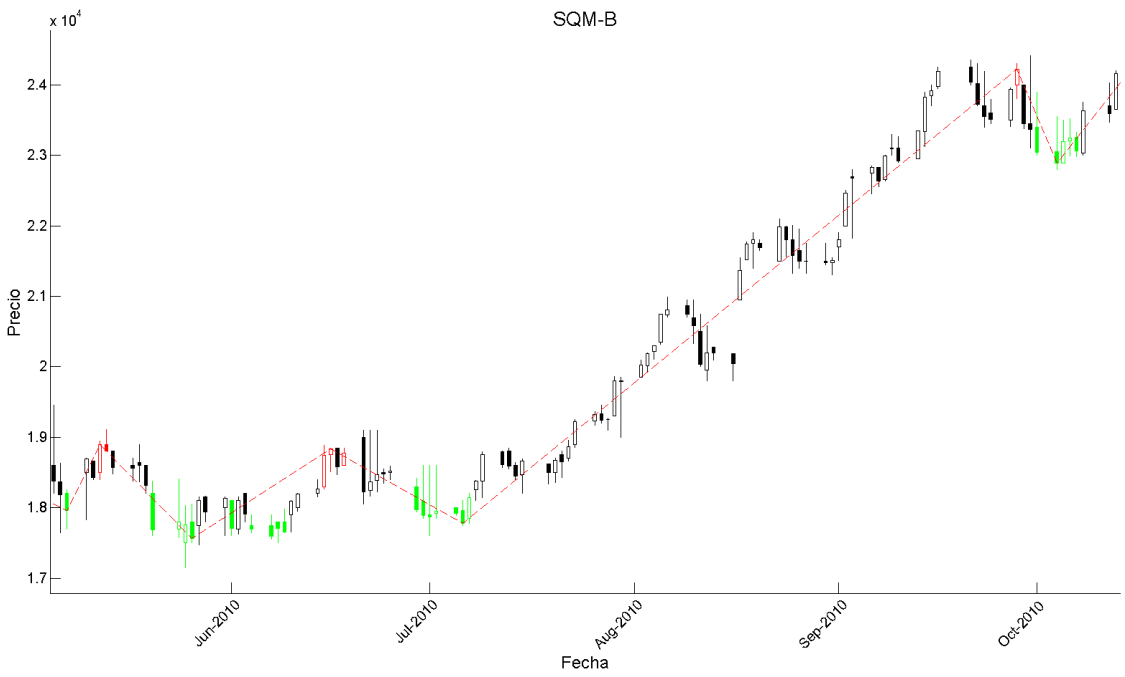

Ilustración 3.7: Extensión temporal de compraventa mostrando buen comportamiento ante retracciones fuera de rango

Se realiza un nuevo análisis de distribución de variables dependientes, los resultados se muestran en la Ilustración 3.8. En ésta se aprecia que se cumple el objetivo de aumentar el número de etiquetas positivas, aumentando ésta de menos de 1% para retracción por sobre 10%, hasta un 6% para la base de compra y un 16% para la base de venta. Finalmente, se aprecia que en todo el dominio se posee mayor cantidad de etiquetas positivas de venta que de compra, efecto que se atribuye a que, al realizar la extensión temporal, el precio tiende a oscilar en un rango cercano al máximo local antes de bajar. Por el contrario, cuando el precio baja, tiende a alejarse en pocos períodos del último mínimo.

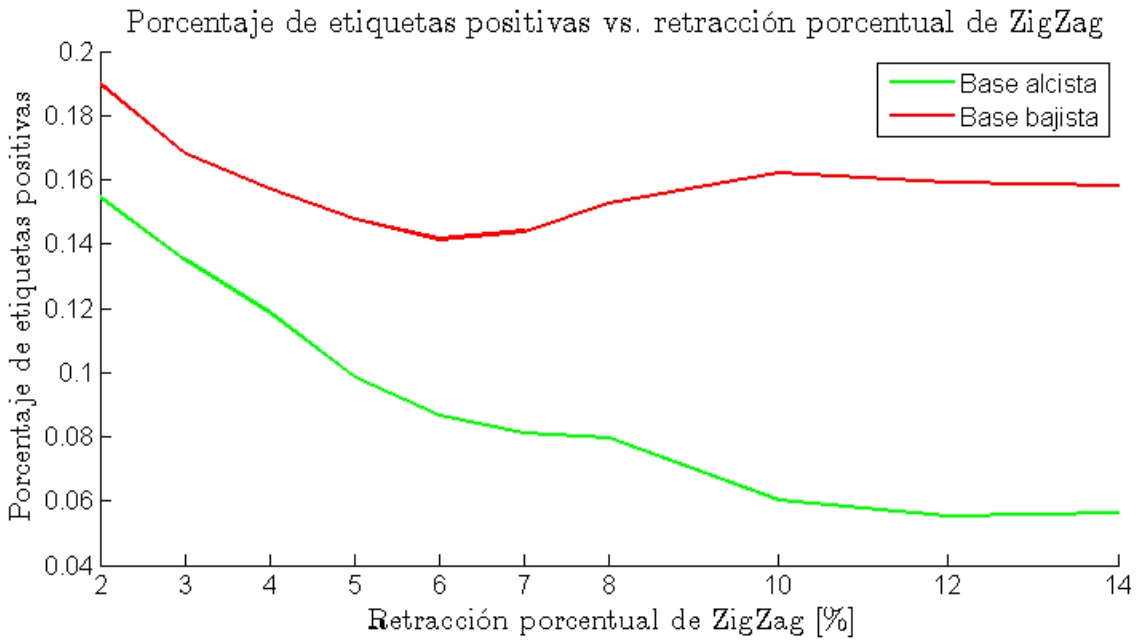

Ilustración 3.8: Distribución de etiquetas positivas a partir de algoritmo de extensión temporal

3.5. Preparación de base de datos

Las bases de datos se deben preparar para ser utilizadas adecuadamente por los modelos de aprendizaje supervisado, éstas se clasifican en: Base de datos en bruto, segmentada y balanceada.

3.5.1. Base de datos en bruto

En principio se posee una base de datos en bruto, ésta se compone de una base de datos para cada una de las 40 acciones que componen el IPSA. Cada base de datos posee el formato expresado en (3.16) y (3.17) para los atributos X y variables dependientes Y respectivamente. En éstas, $\boldsymbol{m}$ representa el número de muestras o períodos de cada acción, $n = 274$ el número de variables independientes y $\boldsymbol{v}$ es el número de variables dependientes con distinto parámetro de retracción de ZigZag.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{n,1} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix} \quad \textbf{(3.16)}$$

$$Y = \begin{bmatrix} y_{1,1} & \cdots & y_{m,v} \\ \vdots & \ddots & \vdots \\ y_{m,1} & \cdots & y_{m,v} \end{bmatrix} \quad \textbf{(3.17)}$$

El parámetro de retracción de ZigZag tiene una relación casi lineal con la distancia temporal de las recomendaciones de compraventa, por lo tango, refleja el ciclo del mercado que se desea predecir. Siendo importante la evaluación de distinto valores para esta variable, evitando sobreajuste por parte de las variables independientes a un ciclo de mercado. Considerando el análisis realizado en 3.3 para la Ilustración 3.1, se decide priorizar valores pequeños de retracción (3.18).

$$PC \in \{0.02 \quad 0.03 \quad 0.04 \quad 0.05 \quad 0.06 \quad 0.07 \quad 0.08 \quad 0.1 \quad 0.12 \quad 0.14\} \quad \textbf{(3.18)}$$

3.5.2. Base de datos segmentada

Como se menciona en la sección 2.4, la base de datos inicial se debe segmentar en entrenamiento, validación y testeo. La base de entrenamiento es utilizada para ajustar los modelos de aprendizaje supervisado, la de validación para encontrar los parámetros óptimos de los modelos evitando sobreajuste y la de testeo completamente independiente hasta la evaluación de resultados finales, es decir, para realizar una simulación realista. Como se menciona en la sección 3.1, sólo se poseen datos transaccionales de Insiders entre 04-01-2010 y 06-05-2013, con el fin de tener suficientes muestras se decide utilizar rango completo de fechas en la segmentación (Tabla 3.2).

Segmentación	Fecha Inicial	Fecha Final	Duración (Aprox.)
Entrenamiento	04-01-2010	01-01-2012	2 años
Validación	02-01-2012	01-01-2013	1 año
Testeo	02-01-2013	03-05-2013	4 meses

Tabla 3.2: Fechas de segmentación de las bases de datos

Se obtienen las tres segmentaciones para cada una de las 40 acciones, sumando un total de 120 bases en la etapa de segmentación. Se crea una base global para cada segmentación, que combina las variables dependientes e independientes de las 40 acciones.

3.5.3. Base de datos balanceada

La correcta utilización de la métrica de *Accuracy* en la fase de entrenamiento requiere utilizar bases de datos con etiquetas balanceadas (ver sección 2.4). Cada etiquetado posee distinto número de valores positivos, por lo que se debe balancear para cada una de las 40 acciones y 20 variables dependientes de extensiones de ZigZag (diez valores para bases del tipo *Long* y *Short*) por separado, sumando un total de 800 bases de datos balanceadas para cada una de sus tres segmentaciones, generando 2400 bases balanceadas en total.

En la fase de minería de datos se desea utilizar las bases balanceadas para entrenamiento y validación. Con el fin hacer el entrenamiento representativo, al igual que en la fase de segmentación, se generan 60 bases globales (tres segmentaciones de las 20 variables dependientes), la cual es representativa de los 40 activos.

Capítulo 4

Aplicación de minería de datos

En esta fase se realiza el proceso de extracción de conocimiento, su fin es predecir períodos de cambio de tendencia, períodos de retracción del precio o mínimos y máximos locales, a partir de las variables independientes ya pre-procesadas y transformadas. Para esto, se utilizan herramientas de aprendizaje supervisado.

En primera instancia se analiza el poder predictivo de los atributos. Para esto, se utiliza selección de características mediante FS, BE y combinaciones de éstos, seleccionando el subconjunto con mayor valor predictivo. A continuación, se aplican los modelos de aprendizaje supervisado no lineales (BPNN, SVM y SBM) sobre la mejor selección de atributos en la base balanceada. Finalmente, con el fin de obtener predicciones que sigan la distribución real de los datos, se evalúan combinaciones de predicciones de modelos ya entrenados sobre la base segmentada no balanceada.

Todos los entrenamientos se realizan para las 20 bases de datos (diez valores de retracción de ZigZag para compra y venta). Cada modelo se ajusta utilizando una base de datos global, la que combina los datos de las 40 acciones, de esta manera, la evaluación del poder predictivo de los atributos extraídos es representativa.

4.1. Selección de características

Se desea determinar el poder predictivo de los atributos extraídos, para esto, en principio se debe considerar que no todos los atributos poseen poder predictivo, pudiendo algunos ingresar ruido al modelo predictor, disminuyendo desempeño.

Se realiza un análisis de selección de características mediante *Forward Selection* y *Backward Elimination*. En éstos se utiliza como modelo la regresión logística, la cual es equivalente al modelo BPNN expuesto en la sección 2.4.3.3 sin capas ocultas y sin regularización ($\lambda = 0$). Se utilizan los diez valores de retracción de ZigZag PC expuestos en la sección (3.18) sobre las bases alcistas y bajistas.

La selección del modelo de regresión logística no da certeza sobre si los atributos seleccionados sean relevantes para un modelo no lineal. En estricto

rigor, se debiese aplicar este proceso para cada modelo, sin embargo, el modelo seleccionado se considera una aproximación válida debido al tiempo disponible.

La selección se realiza mediante la métrica AUC, su objetivo es evaluar de forma genérica a que retracción de ZigZag se ajustan mejor las variables independientes, es decir, qué ciclo del mercado describen mejor los atributos generados.

Se evalúan distintos subconjuntos de atributos; los resultados se muestran en la Ilustración 4.1 para la base alcista e Ilustración 4.2 para la base bajista. En éstas se muestra el AUC para distintas retracciones de ZigZag con los siguientes subconjuntos de variables independientes:

- Todos: Todos los atributos.
- Alcistas: Sólo atributos compra.
- Bajistas: Sólo atributos venta.
- FS: Subconjunto de variables seleccionadas mediante FS.
- BE: Subconjunto de variables seleccionadas mediante BE.
- $FS\&BE$: Subconjunto de FS y BE con operador lógico AND.
- $FS|BE$: Subconjunto de FS y BE con operador OR.

En la Ilustración 4.1 e Ilustración 4.2 se distingue una clara tendencia a obtener mejores resultados predictivos utilizando un valor de retracción de ZigZag pequeño, por lo que las variables independientes predicen mejor cortos intervalos temporales.

En la base alcista se aprecian mejores predicciones con RP en el rango $[3\%, 5\%]$ (entre seis y trece períodos en mediana y entre 10.83 y 20.26 períodos en promedio <Tabla F.>), por otra parte, en la base bajista se aprecia un máximo local en $RP = 3\%$ y un máximo global en $RP = 6\%$.

En ambas bases se distingue que utilizando todas las variables se posee mayor valor predictivo que utilizando sólo los atributos de compra o venta, resultado atribuido a que, eventualmente, puede haber variables independientes de compra o venta que posean valor predictivo sobre la base contraria. En general, si se utilizan atributos sólo de compra o venta, se obtienen mejor resultados con los atributos de compra para la base alcista y de venta para la base bajista, resultado no válido para todas las retracciones, por lo que puede atribuirse a ruido estadístico.

Cabe destacar que se predice mejor utilizando atributos seleccionados mediante FS y BE que con todos los datos, es decir, hay variables que agregan más ruido que valor predictivo, ratificando que la utilización de selección de variables es un mejor enfoque que utilizar todas éstas.

En la Tabla 4.1 se resume el mejor AUC con su retracción de ZigZag respectiva para todos los subconjuntos sobre ambas bases. En ésta se observa que el mejor AUC para la base alcista se encuentra utilizando FS con $RP = 4\%$ y tiene un valor de 0.8259, por otra parte, para la base bajista se obtiene un mejor AUC de 0.7877 utilizando FS con $RP = 6\%$.

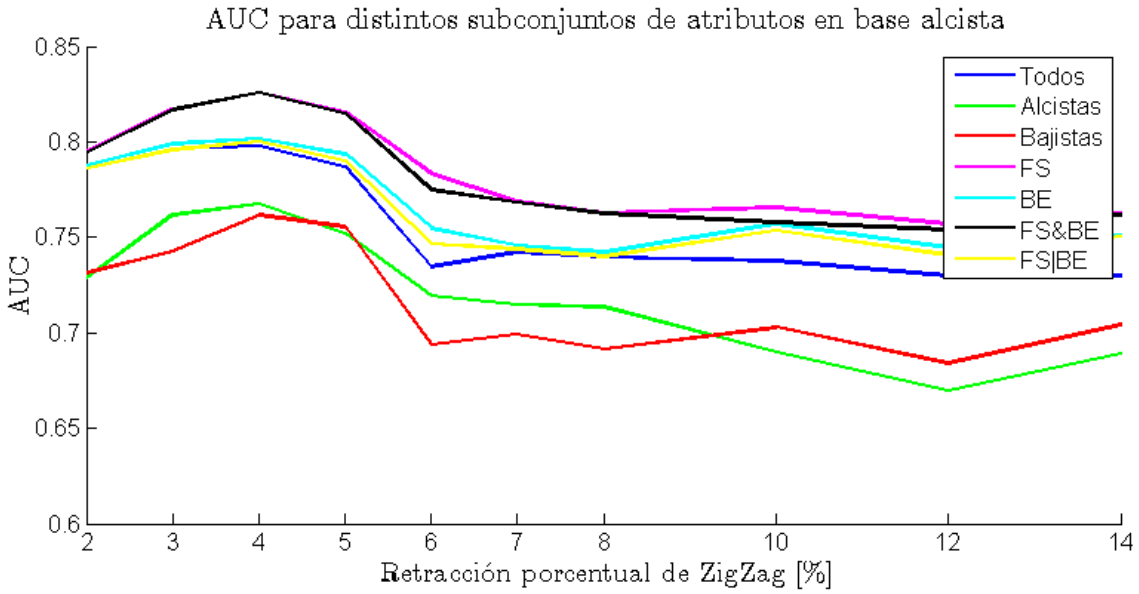

Ilustración 4.1: AUC para distintos subconjuntos de atributos en base alcista

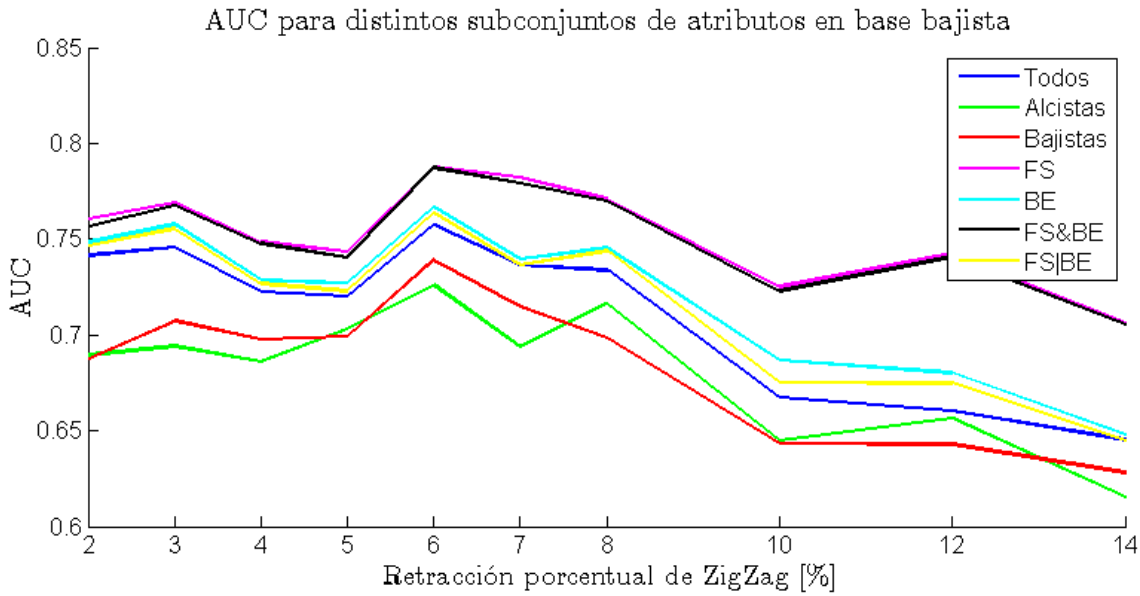

Ilustración 4.2: AUC para distintos subconjuntos de atributos en base bajista

Subconjunto	Base Alcista		Base Bajista	
	Mejor AUC	RP ZigZag	Mejor AUC	RP ZigZag
Todas	0.7978	4%	0.7577	6%
Alcistas	0.7674	4%	0.7261	6%
Bajistas	0.7617	4%	0.7393	6%
FS	0.8259	4%	0.7877	6%
BE	0.8017	4%	0.7558	6%
FS&BE	0.8256	4%	0.7873	6%
FS|BE	0.8001	4%	0.7634	6%

Tabla 4.1: Mejor AUC y RP respectivo para distintos subconjuntos de atributos

4.2. Análisis predictivo de atributos

Se desea analizar por separado el poder predictivo de los atributos generados, para esto se utilizan los subconjuntos seleccionados mediante FS y BE para las distintas retracciones porcentuales de ZigZag. La utilización de FS se basa en que, por construcción, éste selecciona sólo las variables con mayor valor predictivo, por otro lado, BE elimina los atributos que agregan mayor ruido. Las pruebas realizadas corresponden a:

- Evaluar la importancia general de los atributos extraídos, para esto se grafica la selección total de atributos mediante FS y BE en ambas bases (Ilustración 4.3).

- Determinar la distribución de variables importantes y ruidosas para métricas extraídas de Insiders, MACD, RSI y SO (Ilustración 4.4).

- Analizar la importancia de los distintos tipos de atributos generados a partir de indicadores técnicos, diferenciando divergencia, cruce y agotamiento. (Ilustración 4.5).

- Examinar la importancia de las distintas configuraciones de divergencia (Ilustración 4.6).

En la Ilustración 4.3 se observa el porcentaje de variables seleccionadas mediante FS y BE para las bases alcistas y bajistas. En ésta, se aprecia que FS realiza una selección minuciosa de pocas variables (las con mayor valor predictivo), siendo las importantes en el rango de 6% a 15% (entre 18 y 43 atributos). Por otra parte, BE en general selecciona casi todas las variables, nunca eliminando más del 10% de los atributos. Es decir que, en general, el 15% de las variables poseen poder predictivo importante y alrededor del 5% agrega más ruido que valor predictivo. De la selección mediante BE se desprende que para mayores retracciones de ZigZag, se tienen menos variables que agregan ruido en la venta, sucediendo lo contrario en compra.

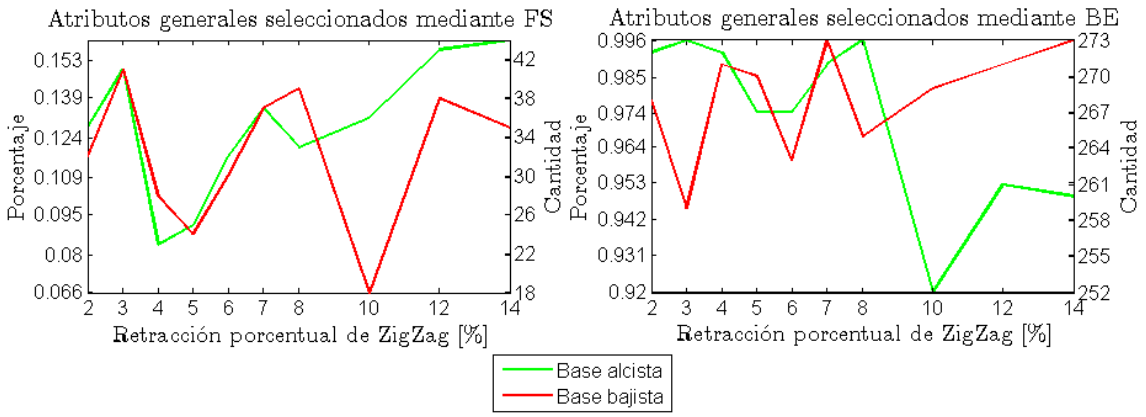

Ilustración 4.3: Distribución de selección de características totales utilizando FS y BE

En la Ilustración 4.4 se grafican los porcentajes de características seleccionadas mediante FS y BE para variables extraídas de Insider, MACD, RSI y SO. En la parte superior se aprecia que FS, en general, selecciona como más importantes las variables de Insider para venta a mediano plazo, no siendo útiles a corto ni largo plazo, lo que se puede atribuir a que los atributos no poseen valor predictivo u otra variable agrega el mismo tipo de información. Por el contrario, para compra siempre son de utilidad, excepto a muy corto plazo. La selección mediante BE muestra que las variables de Insider de recomendación de venta casi no agregan ruido en la predicción, por el contrario, las de compra tienden a agregar ruido, sobre todo para plazos mayores.

En MACD se observa que se tienden a elegir como importantes (mediante FS) mayor cantidad de variables para predecir venta a muy corto, mediano y largo plazo (forma de "W"), esto se atribuye a que los parámetros utilizados en su cálculo se adaptan mejor a estos ciclos del mercado para venta. Por otro lado, para compra se percibe una selección casi equidistribuida de atributos, con excepción para corto plazo en el mismo rango que no se eligen variables para venta. En la selección mediante BE se obtiene que, en general para corto plazo, se tiene la misma cantidad de variables eliminadas para compra y venta (menor al 3%), por lo que no hay muchas variables que agreguen ruido. Cabe destacar que para mayores plazos las variables de compra tienden a agregar más ruido (no predicen en fase con los ciclos mayores), efecto contrario de las variables de venta.

En cuanto a los atributos seleccionados de RSI, se infiere claramente que se seleccionan con mayor valor predictivo más variables de compra que de venta, con muy pocos atributos de venta que agreguen valor predictivo para largo plazo. En la selección mediante BE se aprecia que, en ambas bases, casi siempre se eliminan uno o dos atributos (excepto para $RP \in \{4\%, 7\%\}$), por lo que RSI conviene sea utilizando en este rango, ya que agrega menor ruido predictivo.

Finalmente, se aprecia que las variables de SO agregan (en ambas bases) mayor valor predictivo a muy corto plazo, luego su desempeño decae y tiende a aumentar en ciclos más largos. En la selección utilizando BE se observa que las variables de venta tienden a agregar mayor ruido a corto plazo (decayendo a plazos mayores), teniendo el efecto contrario para predecir compra. Cabe destacar que utilizando $RP = 7\%$ es cuando las variables de SO agregan menos ruido.

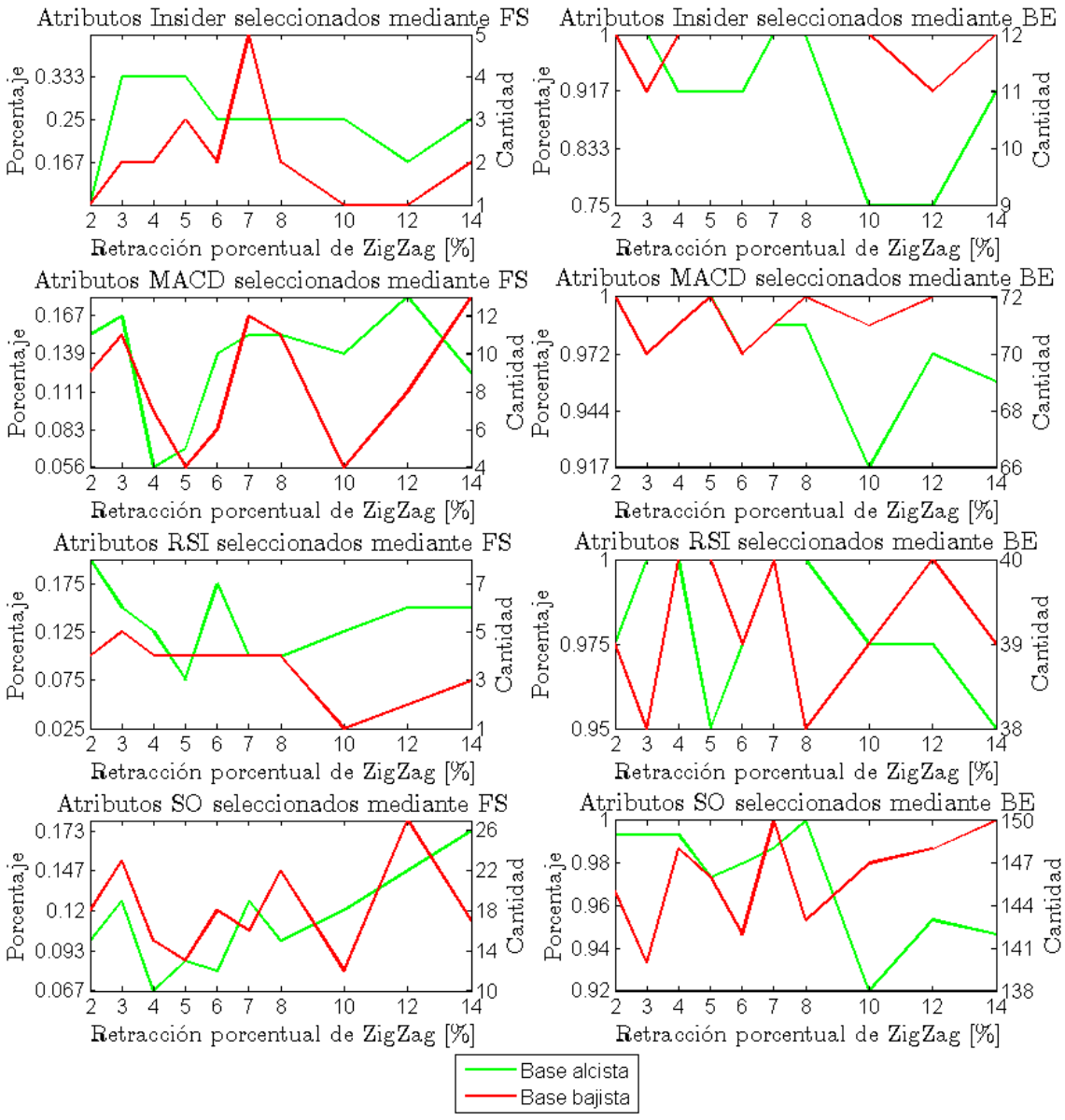

Ilustración 4.4: Porcentaje de Selección de características por Indicador con FS y BE

En la Ilustración 4.5 se grafica el porcentaje de tipo de variables (divergencia, cruce y agotamiento) mediante FS y BE para las bases alcistas y bajistas. Se observa que las divergencias para compra y venta tienen casi la misma cantidad de variables predictivas en todo el dominio de ciclos analizados. La selección mediante BE deja claro que para compra agrega poco ruido a cortos plazos y tiende a aumentar a largo plazo, sucediendo el efecto contrario para predecir venta.

En las variables de cruce, se aprecia en ambas bases un mayor valor predictivo a muy corto, mediano y largo plazo (forma de "W"). Además, tienden a haber mayor cantidad de atributos que agregan ruido para predecir venta.

Finalmente, en las variables de agotamiento se manifiesta mayor valor predictivo a muy corto plazo, casi ninguno a corto-mediano plazo y mayor importancia a largo plazo. Además, en la selección mediante BE los atributos tienden a agregar mayor ruido para predecir compra a largo plazo y venta a corto plazo.

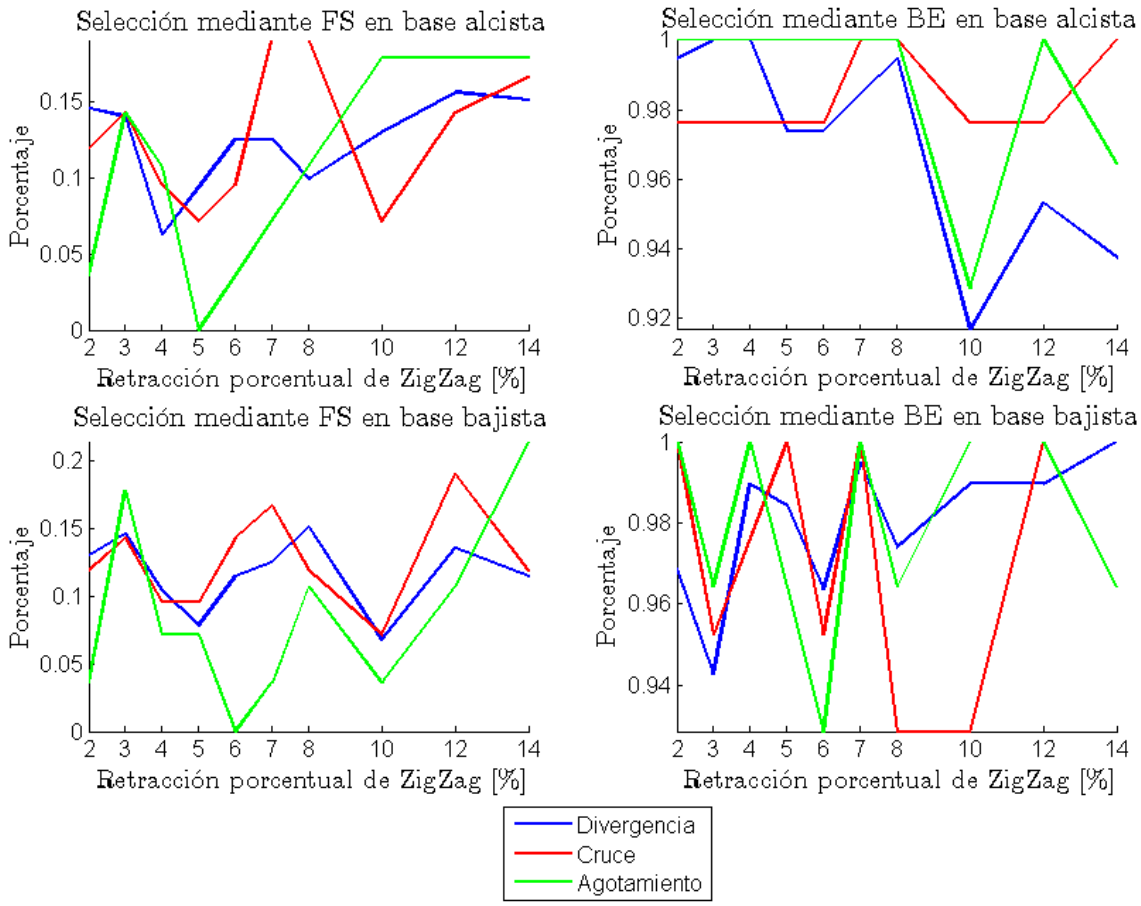

Ilustración 4.5: Porcentaje de Selección de características por tipo de variable con FS y BE

En la Ilustración 4.6 se muestra el porcentaje atributos seleccionados mediante FS y BE para distintas configuraciones de divergencia. Se aprecia que para determinar compra se obtienen pocas variables con valor predictivo en $RP = 4\%$, aumentando gradualmente para mayores plazos. Por el contrario, para venta casi siempre se tienen variables con valor predictivo, observándose máximos locales en $RP \in \{3\%, 8\%, 12\%\}$. En general, se destaca como mejor combinación la utilización de divergencias con patrón regular normal (no triple) y confirmación doble, teniendo el mismo resultado para casi todo el dominio.

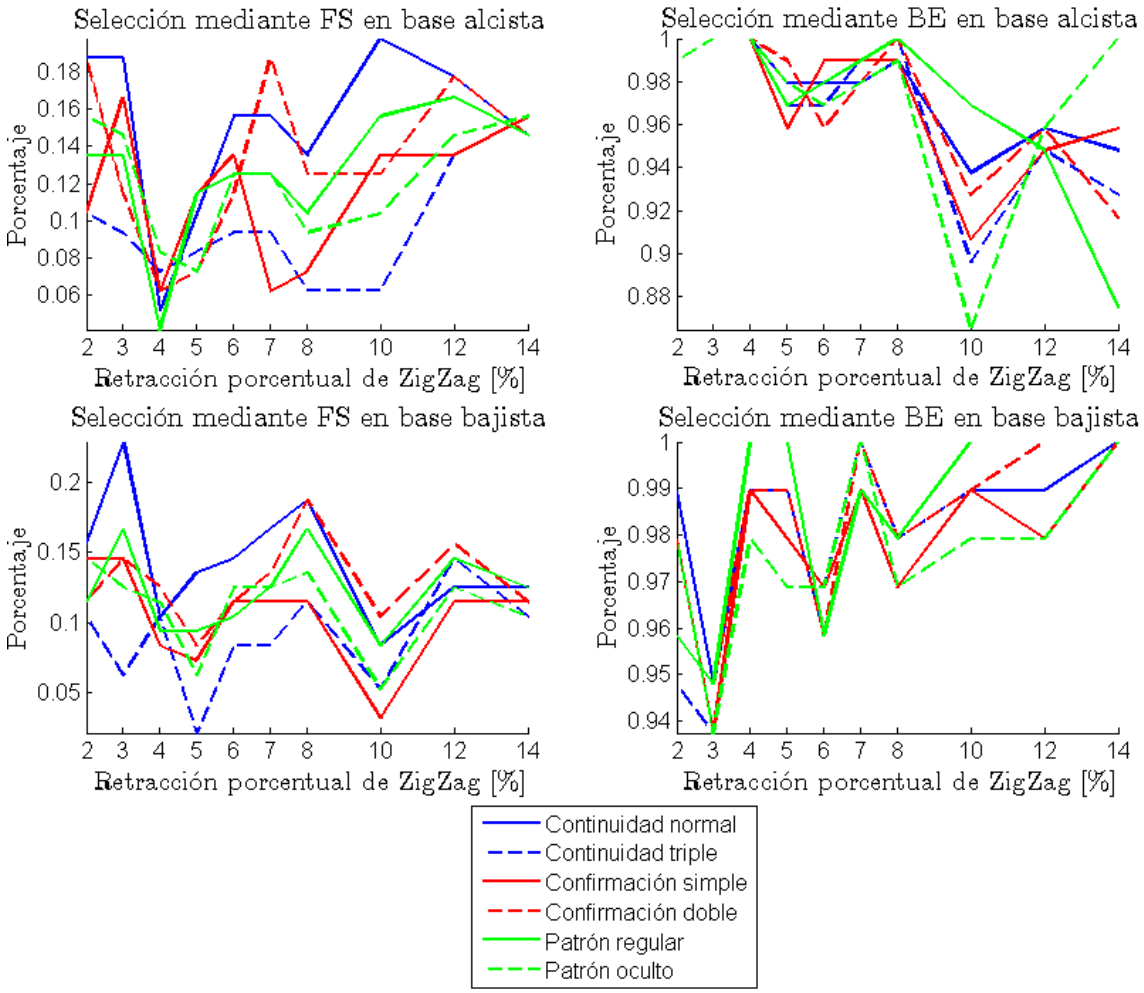

Ilustración 4.6: Porcentaje de selección de características por tipo de divergencia con FS y BE

4.3. Entrenamiento sobre base balanceada

Se realizan entrenamientos utilizando BPNN, SVM y SBM sobre las bases balanceadas; el objetivo es determinar los parámetros que se ajustan mejor a los distintos ciclos de mercado para cada modelo. La razón de utilizar modelos elaborados con capacidad de extraer estructura altamente no lineal, es la de obtener buen desempeño al momento de predecir posibles cambios de tendencia de un activo.

Para cada entrenamiento se utiliza el mejor subconjunto de atributos encontrado en la sección 4.1 para cada valor de retracción porcentual de ZigZag. Se evalúa el desempeño utilizando las métricas AUC (apropiada para comparar modelos) y *Accuracy* (apropiada para determinar mejor modelo predictor en bases balanceadas).

Todos los entrenamientos se realizan sobre las bases alcistas y bajistas para los diez parámetros de retracción porcentual de ZigZag expresados en la sección (3.18). Con el fin de obtener resultados representativos se utiliza la base global (combinación de 40 activos). Esta decisión se respalda debido a la restricción de rango de tiempo en que se poseen datos de Insiders, por lo que no se tendrían datos suficientes para entrenar efectivamente los modelos para cada activo por separado.

En el entrenamiento utilizando BPNN se fija el número de capas ocultas en uno debido a restricciones de tiempo. Se utilizan las combinaciones de conjuntos de parámetros para número de neuronas y regularización mostradas en (4.1) y (4.2) respectivamente. Con el objetivo de evitar mínimos locales, se realizan cuatro entrenamientos por set de parámetros, resultando en 468 $(13 \cdot 9 \cdot 4)$ entrenamientos de BPNN por base de datos, resultando en un total de 9.360 $(20 \cdot 468)$ entrenamientos.

$$\#Neuronas \in \{10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70\} \quad \textbf{(4.1)}$$

$$\lambda \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10\} \quad \textbf{(4.2)}$$

El entrenamiento con SVM se realiza con kernel Gaussiano con el fin de extraer patrones altamente no lineales. La utilización apropiada de esta herramienta implica variar el parámetro de ancho de banda del kernel σ_{SVM} y el parámetro de margen C. En particular se utilizan las combinaciones de los valores expuestos en (4.3) y (4.4), resultando en 132 $(12 \cdot 11)$ entrenamientos por base de datos. Dado que se poseen 20 bases de datos, se realiza un total de 2640 $(132 \cdot 20)$ entrenamientos.

$$\sigma_{SVM} \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 200\} \quad \textbf{(4.3)}$$

$$C \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100\} \quad \textbf{(4.4)}$$

Para el entrenamiento con SBM se aplica kernel Gaussiano por la misma razón mencionada previamente, por lo que su utilización adecuada involucra variar el parámetro de ancho de kernel σ_{SBM}. En particular se utilizan los valores expresados en (4.5), resultando en trece entrenamientos de SBM por base de datos. Como se debe entrenar para las 20 bases, se realiza un total de 300 $(15 \cdot 20)$ entrenamientos.

$$\sigma_{SBM} \in \{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 500, 700, 1000\} \quad \textbf{(4.5)}$$

En la Ilustración 4.7 e Ilustración 4.8 se grafica el mejor AUC y *Accuracy* obtenido por los modelos para las distintas retracciones porcentuales de ZigZag sobre la base de validación global balanceada. Cabe destacar que en ambas ilustraciones se obtienen resultados de AUC con distribución muy parecida a la obtenida en la selección de características (Ilustración 4.1 e Ilustración 4.2).

En la Ilustración 4.7 se aprecia para la base alcista que todos los modelos obtienen el mejor desempeño (con AUC y *Accuracy*) en $RP = 4\%$. El mejor rendimiento global se logra con BPNN con valores de $\mathrm{AUC} = 0.826$ y $Accuracy = 0.7601$. Finalmente, se reconoce que los tres modelos poseen un comportamiento muy parecido para ambas métricas en el dominio evaluado, con una clara tendencia a disminuir a mayores ciclos de mercado.

En la Ilustración 4.8 se observa para la base bajista que el mejor resultado (con ambas métricas) se obtiene con todos los modelos en $RP = 6\%$. Al igual que con la base alcista, se obtienen resultados muy parecidos en AUC y *Accuracy* en todo el dominio con los tres modelos, siendo SVM relativamente inferior. Los mejores desempeños se obtienen con el modelo BPNN con valores de $\mathrm{AUC} = 0.7855$ y $Accuracy = 0.7244$.

En la Tabla 4.2 y Tabla 4.3 se muestran los parámetros de los modelos con mejor *Accuracy* para la base de alcista y bajista respectivamente. En ésta se aprecia que los parámetros del mejor modelo obtenido no se encuentran en los extremos de los valores evaluados, por lo que el entrenamiento es realizado de forma adecuada.

Para mejorar los resultados obtenidos se puede utilizar otro modelo de aprendizaje supervisado, afinar la grilla de parámetros de entrenamiento de los modelos, utilizar más datos, generar más variables con valor predictivo o utilizar una mejor selección de atributos.

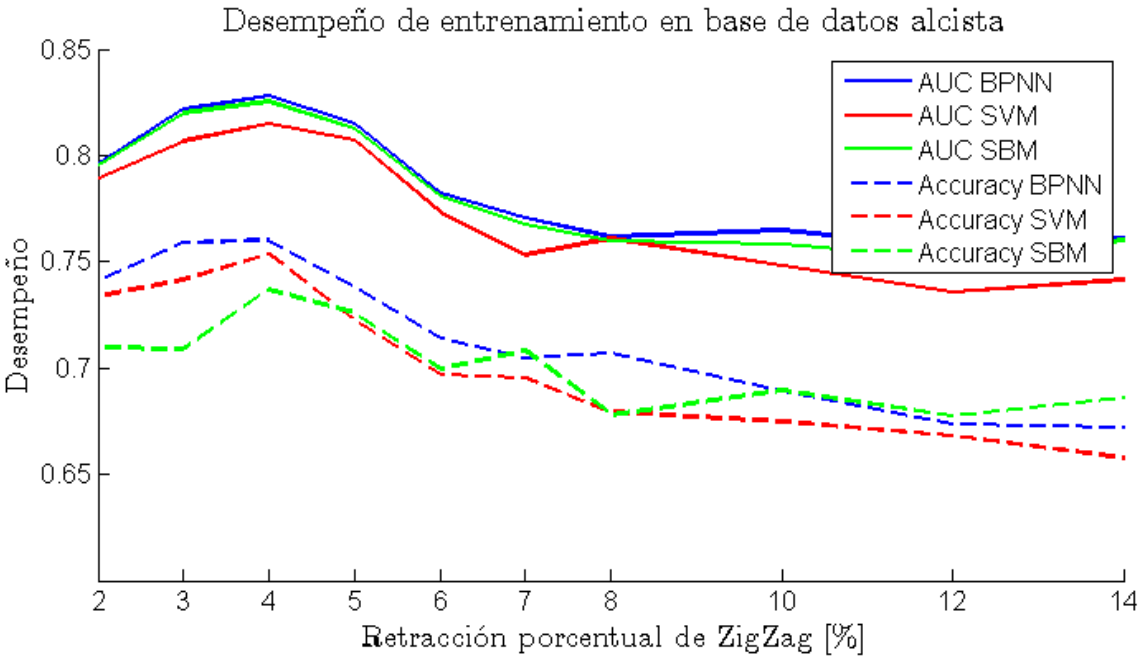

Ilustración 4.7: Mejor resultado de aprendizaje supervisado sobre base de validación alcista

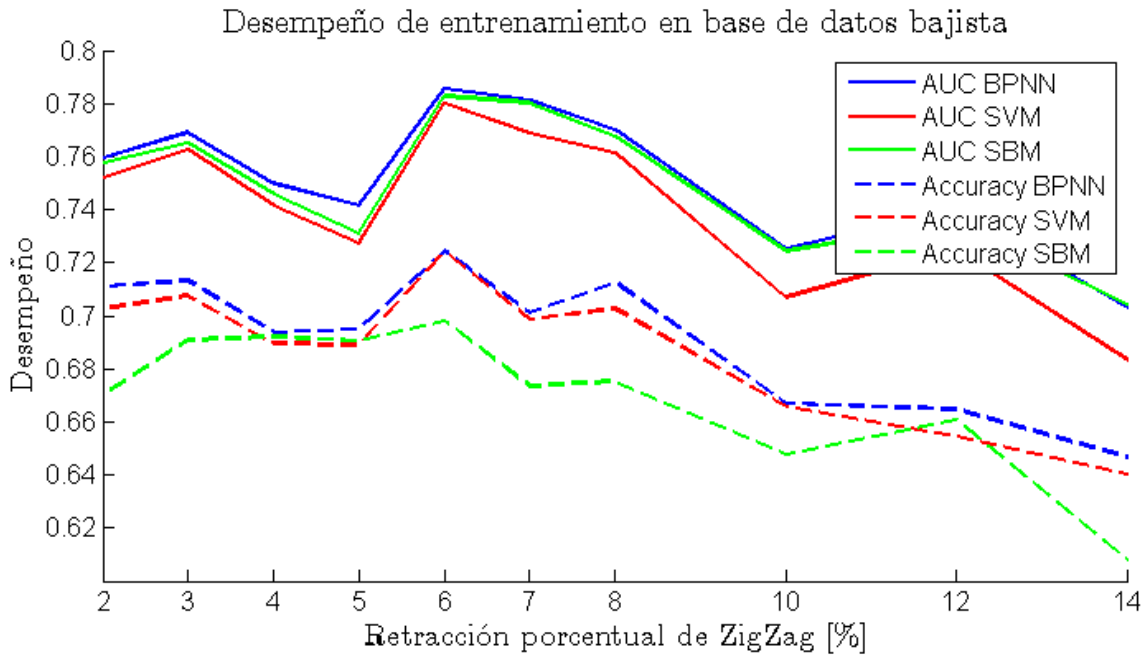

Ilustración 4.8: Mejor resultado de aprendizaje supervisado sobre base de validación bajista

Modelo	*Accuracy*	*RP* ZigZag	Parámetros
BPNN	0.7601	4%	$\#\text{Neuronas} = 60, \quad \lambda = 0.3$
SVM	0.7535	4%	$\sigma_{SVM} = 30, \quad C = 3$
SBM	0.737	4%	$\sigma_{SBM} = 0.1, \quad Th = 0.418$

Tabla 4.2: Mejores resultados sobre base alcista de validación (balanceada)

Modelo	*Accuracy*	*RP* ZigZag	Parámetros
BPNN	0.7244	6%	$\#\text{Neuronas} = 35, \quad \lambda = 0.1$
SVM	0.724	6%	$\sigma_{SVM} = 3, \quad C = 0.03$
SBM	0.6981	6%	$\sigma_{SBM} = 300, \quad Th = 0.3988$

Tabla 4.3: Mejores resultados sobre base alcista de validación (balanceada)

Se desea determinar el comportamiento de los modelos predictores entrenados para cambios de tendencia futuros, para esto, se predice sobre las bases de validación utilizando distintas métricas. Las predicciones son realizadas utilizando el mejor subconjunto de atributos seleccionados para las retracciones de ZigZag mostradas en la Tabla 4.2 y Tabla 4.3.

La Ilustración 4.9 muestra la curva ROC para los tres modelos predictores sobre las bases de cambios de tendencia alcistas y bajistas. En ésta se aprecia claramente que la curva ROC sobrepasa la predicción aleatoria, por lo que se puede inferir que los atributos seleccionados logran satisfactoriamente capturar estructura predictiva.

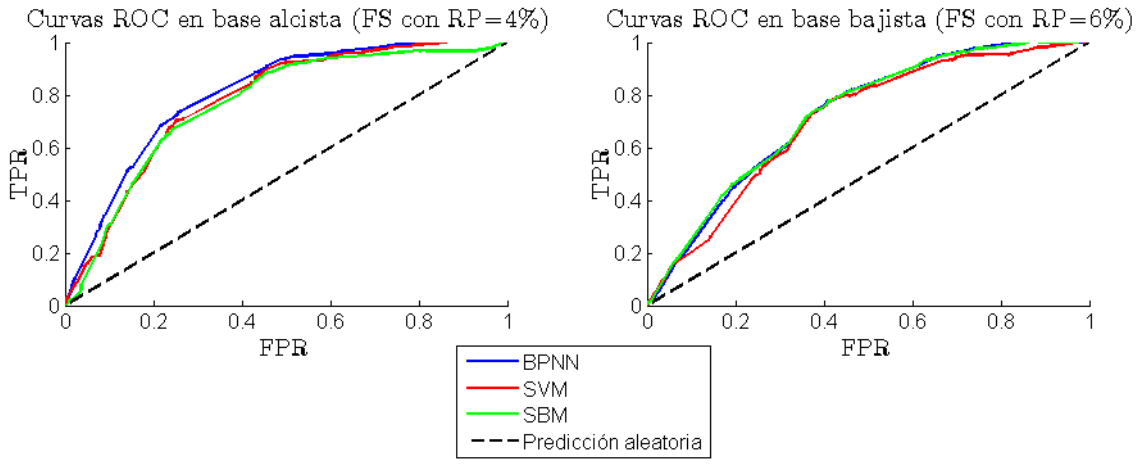

Ilustración 4.9: Cruda ROC para predicción con mejor RP de ZigZag y selección de características

En la Ilustración 4.10 se aprecian los desempeños obtenidos para las métricas de AUC, *Accuracy*, *Precision*, *Recall* y *F1-Score*. Pese a que los resultados son muy parecidos para los tres modelos, cada uno presenta distintas características predictivas, motivo por el cual se debe realizar la evaluación del comportamiento de los modelos (o combinación de éstos) sobre las bases segmentadas no balanceadas y así utilizar el que presente mejor rendimiento general. Finalmente, en Anexo G se presentan las matrices de confusión para los modelos entrenados.

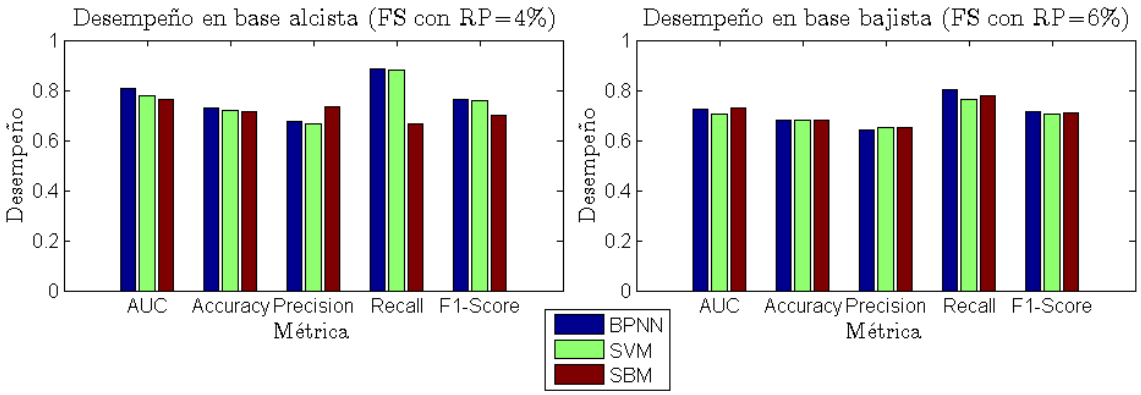

Ilustración 4.10: Métricas de predicción para mejor *Accuracy* con mejor retracción porcentual de ZigZag y subconjunto de atributos

4.4. Evaluación sobre base no balaceada

Se realiza una fase de evaluación de los modelos predictores entrenados con las bases balanceadas. Ésta se realiza prediciendo sobre las bases alcistas y bajistas de validación segmentadas, pero no balanceadas.

Dado que las bases poseen la distribución de etiquetas positivas mostrada en la Ilustración 3.8, éstas no se encuentras balanceadas, por lo que la utilización de *Accuracy* como métrica no es apropiada, seleccionando AUC como medida de desempeño.

Con el objetivo de no perder generalidad, además de analizar todos los modelos con las retracciones de ZigZag previamente evaluadas, se genera una nueva simulación. Ésta se basa en utilizar cada modelo con su retracción óptima y no fijar un RP arbitrario, de esta forma cada modelo predice el ciclo al que mejor se adaptan los atributos según la Tabla 4.2 y Tabla 4.3 para alza y baja respectivamente.

En el análisis de los modelos propuestos y combinaciones de éstos, se utilizan los modelo BPNN, SVM y SBM por sí solos (modelos uno a tres), su combinatoria utilizando redes Bayesianas ingenuas (modelos cuatro a siete) y el operador lógico “AND” (modelos nueve a doce). Se considera un caso especial, el cual consiste en aplicar operador lógico “AND” a los tres modelos predictores por sí solos (modelo cuatro) y al resultado de combinar éstos mediante NB (modelo ocho). No se realizan todas las combinaciones posibles al operador lógico, ya que éstas corresponden a la combinación sin repetición de los primeros siete modelos, de acuerdo a la ecuación (4.6) éstas son 127 combinaciones.

$$\#Combinaciones(\#Casos) = \sum_{k=1}^{\#Casos} \binom{\#Casos}{k} \qquad \textbf{(4.6)}$$

Finalmente, los modelos evaluados se resumen en:

1. BPNN: Red Neuronal de Retropropagación.
2. SVM: Maquina de Soporte Vectorial.
3. SBM: Métodos Basados en Similitud.
4. $NB(1,2,3)$: Red Bayesiana que combina resultados de modelos 1,2 y 3.
5. $NB(1,2)$: Red Bayesiana que combina resultados de modelos 1 y 2.
6. $NB(1,3)$: Red Bayesiana que combina resultados de modelos 1 y 3.
7. $NB(2,3)$: Red Bayesiana que combina resultados de modelos 2 y 3.
8. $\&(1,2,3,4)$: Operador lógico AND que combina resultados 1 a 4.
9. $\&(1,2,3)$: Operador lógico AND que combina resultados 1 y 2.
10. $\&(1,2)$: Operador lógico AND que combina resultados 1 y 2.
11. $\&(1,3)$: Operador lógico AND que combina resultados 1 y 3.
12. $\&(2,3)$: Operador lógico AND que combina resultados 2 y 3.

El modelamiento posterior con redes Bayesianas ingenuas se realiza con la intención de que este modelo extraiga la proporción real de la variable objetivo. Para evitar sobreajuste, los modelos NB son entrenados con los primeros dos tercios de los períodos de la base de validación no balanceada y evaluado su desempeño con el resto de éstas. Por otra parte, el modelamiento con operadores lógicos se realiza debido a la gran cantidad de valores positivos entregados por lo modelos. El operador “AND” es equivalente a tener múltiples confirmaciones y así aumentar la métrica de exhaustividad o *recall*.

En el modelo combinado ocho los predictores se aplican al mejor valor de retracción de ZigZag (para cada modelo). En general, no todos los modelos poseen el mismo valor de retracción en su óptimo, por lo que se aplica el operador lógico “OR” entre las etiquetas de las variables objetivo de las respectivas RP (para las bases alcista y bajista por separado). De esta manera,

se poseen etiquetas positivas que representan mejor los ciclos de la combinación de modelos.

En la Ilustración 4.11 se presenta el AUC para los modelos evaluados en todo el dominio de parámetros de retracción de ZigZag sobre la base de validación no segmentada, además en $RP = 0$ se encuentra el resultado de mejor retracción para cada modelo.

En general se aprecia que casi todos los modelos se sobreponen en el mismo valor de AUC en todo el dominio (BPNN levemente mayor en base alcista), haciendo difícil la selección del modelo más apropiado. Por otra parte, en ambas bases se distingue una clara tendencia a que los modelos tengan mejor AUC en $PC = 3\%$, por lo que se escoge este valor como adecuado.

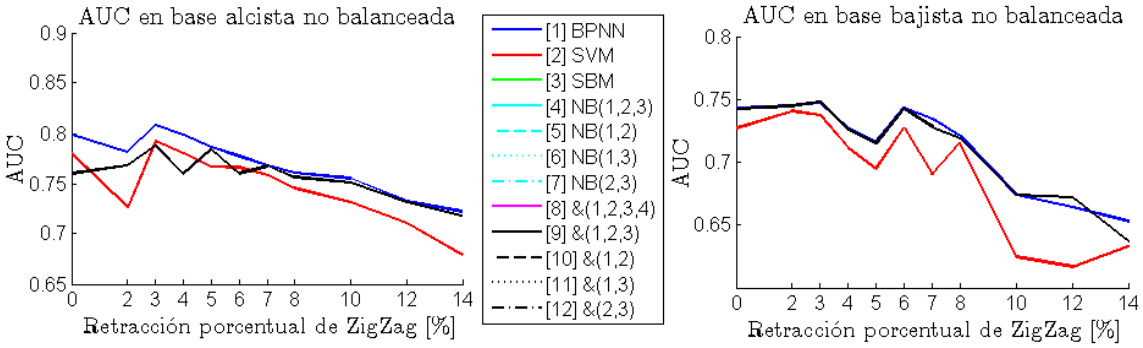

Ilustración 4.11: AUC para modelos evaluados sobre base no balanceada

Debido a que la métrica AUC no permite discernir adecuadamente entre los modelos propuestos, se realiza un análisis de distintas métricas para ambas bases con $RP = 3\%$. Entre las métricas de evaluación se considera: *Accuracy*, *F1-Score* (combina *Precision* y *Recall*), Diferencia Porcentual (*Percentual Difference* <PD>) y la Media Harmónica de todas las anteriores (*Overall Harmonic Mean* <OHM>).

La métrica PD representa el porcentaje total de error de valores señalados de clase positiva, con respecto a la cantidad real de valores que pertenecen a esa clase. Su fin es discernir si un modelo predice con una distribución porcentual distinta a la de la base entrenada, su valor es uno si posee la misma cantidad de etiquetados positivos y disminuye mientras más alejado sea la proporción. En función de la matriz de confusión su ecuación se expresa en (4.7).

$$PD = 1 - \frac{|(TP + FP) - (TP - FN)|}{TP + FP + TN + FN} = 1 - \frac{|TP - FN|}{TP + FP + TN + FN} \quad \textbf{(4.7)}$$

La selección final de modelo se realiza mediante la métrica OHM, ésta se expresa en (4.8) y corresponde a la media harmónica entre: AUC, *Accuracy*, *F1-Score* y PD.

$$OHM = 4 \cdot \frac{AUC \cdot Accuracy \cdot F1Score \cdot PD}{AUC + Accuracy + F1Score + PD} \quad \textbf{(4.8)}$$

El resultado de OHM se muestra en la Ilustración 4.12 para los distintos modelos sobre la base alcista y bajista. En cuanto a la base alcista, se aprecia a simple vista que los modelos con mejor OHM son el uno, nueve y diez, sin embargo, al analizar en detalle se percata que el modelo nueve posee OHM levemente mayor a los otros modelos, con un valor de 0.28037. A modo de conclusión, el mejor predictor para la base alcista a partir de las métricas extraídas, es aplicar el operador lógico "AND" entre los resultados de BPNN, SVM y SBM con $PC = 3\%$, es decir, una confirmación de los tres modelos a la vez para cada período (disminuye falsos positivos).

En la base bajista se aprecia que los modelos cinco a siete no poseen *F1-Score* (valores indeterminados), por lo que no son considerados en la elección de modelo. A simple vista los modelos con mejor OHM son el cuatro y ocho, al analizar los valores en detalle se determina que el modelo con mejor OHM es el cuatro con un valor de 0.215375. A modo de resumen, el mejor predictor de venta se obtiene con $RP = 3\%$ y corresponde al modelo NB que utiliza como argumentos las predicciones mediante BPNN, SVM y SBM.

En la Tabla 4.4 se detalla el resultado de los valores de OHM obtenidos. Por otra parte, en el Anexo H se ilustran los resultados para las métricas utilizadas en el cálculo de OHM en ambas bases. Cabe destacar que sobre la base bajista el modelo NB captura de mejor manera la distribución real de etiquetas positivas (mejores métricas de *Accuracy* y PD), motivo por el cual su OHM es mayor y, por lo tanto, seleccionado como modelo apropiado.

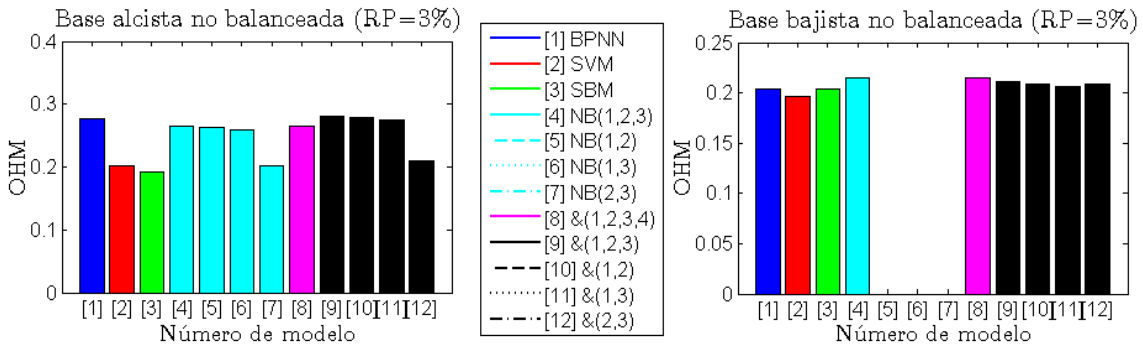

Ilustración 4.12: OHM para modelos evaluados sobre base no balanceada con RP=3%

Base	Número de Modelo					
	1	**2**	**3**	**4**	**5**	**6**
Alcista	0.275897	0.20114	0.192954	0.264659	0.263542	0.259488
Bajista	0.203749	0.195856	0.203306	0.215375	-	-
	7	**8**	**9**	**10**	**11**	**12**
Alcista	0.201838	0.264659	0.28037	0.278629	0.274897	0.209558
Bajista	-	0.215375	0.211611	0.208457	0.206765	0.208952

Tabla 4.4: OHM de modelos evaluados sobre base de validación no balanceada

A modo de resumen, entre los modelos analizados para predecir sobre la base no balanceada, se decide que el mejor modelo para determinar cambios de tendencia hacia el alza es el $\&(1,2,3)$ y hacia la baja $NB(1,2,3)$. Destacando que para ambas predicciones se obtiene mejor resultado combinando los tres modelos predictores.

Capítulo 5

Análisis de predicciones de compraventa

En los capítulos anteriores se ha desarrollado metodológicamente un sistema de que anticipa cambios de tendencia, o equivalentemente, realiza recomendaciones de compraventa de un activo.

En éste capítulo se evalúa cualitativa y cuantitativamente el comportamiento de las predicciones de compraventa. En primera instancia se realiza una evaluación cualitativa, ésta se basa en comparar visualmente el comportamiento a futuro de un activo después de una predicción de cambio de tendencia por parte del sistema.

La evaluación cuantitativa se realiza en base a un simulador de transacciones, que se realiza para la totalidad de los activos y representa los retornos y *drawdown* por operación para cuatro configuraciones específicas.

En todas las evaluaciones se utiliza la base de testeo, ya que las bases de entrenamiento y validación fueron utilizadas previamente para ajustar los modelos, por lo que esta base es completamente independiente de todo lo realizado previamente. De esta forma los ejemplos son representativos; no se sobre ajusta a los datos, ni se pierde generalidad.

5.1. Análisis visual de recomendaciones de compraventa

En el análisis visual se ejemplifican mediante flechas verdes los períodos recomendados para comprar (posible cambio de tendencia alcista) y con flechas rojas períodos de venta (posible cambio de tendencia bajista).

La Ilustración 5.1 ejemplifica las recomendaciones de compraventa para CORPBANCA. En ésta se aprecia que el precio reacciona causalmente a las predicciones, es decir, en la mayoría de los casos el precio varía en dirección favorable a la señal a los pocos períodos después de señalado. Sin embargo, como era de esperarse, también se encuentran predicciones erróneas, razón por la cual es recomendable evaluar cuantitativamente el predictor a través de un modelo de transacción empírico, el cual sea representativo de posibles ganancias y riesgos si se opera directamente con las recomendaciones.

En el Anexo I se ejemplifican las predicciones aplicadas a las acciones de CCU, CFR, CHILE y PARAUCO. En éstas también se aprecia en la mayoría

de los casos que el precio cambia a favor de las recomendaciones de compraventa, por otra parte, se advierte que en CCU no funciona adecuadamente cuando tiene una tendencia alcista pronunciada (mayo 2013), en conjunto con el buen comportamiento observado en CFR y PARAUCO en períodos sin tendencia. De lo anterior se puede inferir que las recomendaciones de compraventa se comportan mejor en períodos de oscilación, por lo que para mejorar resultados se puede combinar el sistema con un filtro de tendencia, realizando operaciones de compraventa sólo en las fases de acumulación o redistribución.

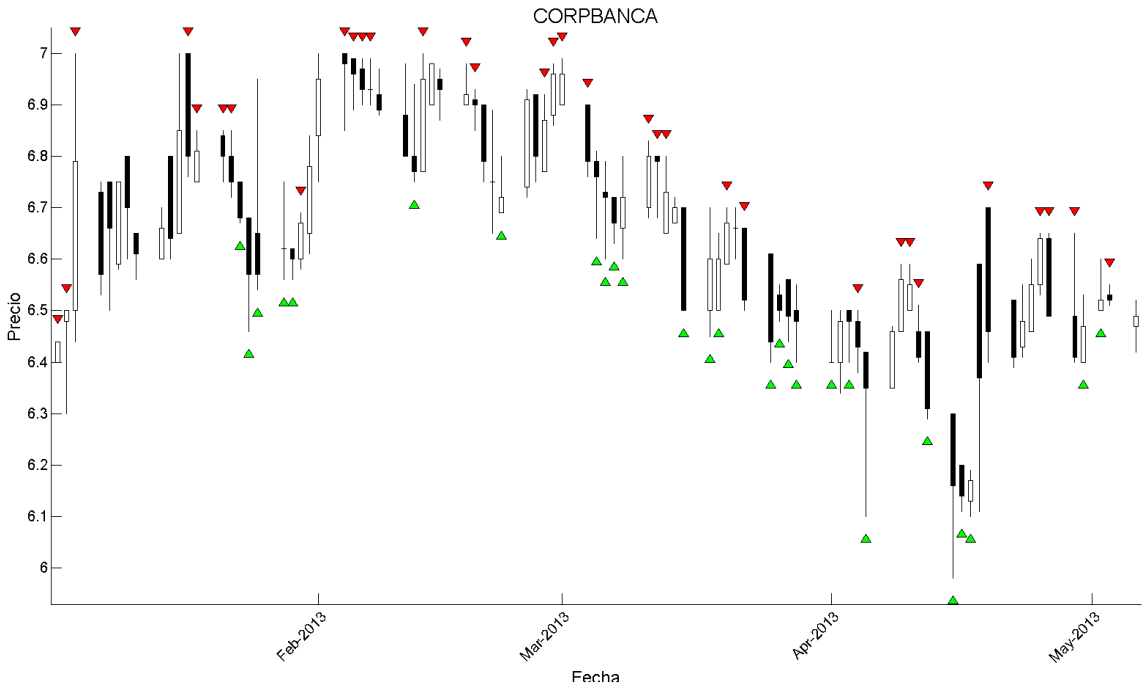

Ilustración 5.1: Resultados predicción aplicado a SQM-B (períodos de testeo)

5.2. Análisis cuantitativo mediante simulación

Se desea determinar cuantitativamente si las predicciones de compraventa poseen utilidad real, para esto, se evalúa de forma realista mediante un simulador de transacciones. Con el fin de evaluar las posibles ganancias y riesgo del sistema, se utilizan las métricas de retorno y *drawdown* por transacción, además, para analizar la duración de las operaciones, se muestra la distribución de ésta para las métricas señaladas.

El simulador propuesto se aplica a las 40 acciones en estudio sobre su período de testeo. Éste considera los siguientes parámetros:

- $windowSize$: Número de períodos hacia el pasado en que se considera $ammountOpenPosition$ y $ammountClosePosition$.
- $ammountOpenPosition$: Número mínimo de recomendaciones en los últimos $windowSize$ períodos para abrir una operación.
- $ammountClosePosition$: Número mínimo de recomendaciones en los últimos $windowSize$ períodos para cerrar una operación. Un valor de uno es que se cierra operación cada vez que hay una señal del tipo contrario, por otro lado, Inf es que no se cierra por heurística.
- $maximumUtilizationOpen$: Máximo número de veces que se puede utilizar una misma recomendación de compraventa para abrir una operación antes que se desactive. Inf es que nunca se desactiva.
- $maximumUtilizationClose$: Máximo número de veces que se puede utilizar una misma recomendación de compraventa para cerrar una operación antes que se desactive. Inf es que nunca se desactiva.
- $maximumPositions$: Máximo número de posiciones abiertas a la vez (para alza y baja por separado). Inf es que no se limita.
- $hedging$: variable binaria que representa si se pueden abrir operaciones de tipo alcista y bajista vez.
- $numberPositionsToClose$: Máximo número de posiciones que se cierran a la vez por heurística de cierre.
- $instantClose$: Variable binaria que representa si los *stop-loss* y *take-profit* se aplican al precio de cierre o en el momento de cruce del precio.
- $stopLossLong$: Umbral de s*top-loss* porcentual estático aplicado para las operaciones del tipo alcista. Inf es no utilizar *stop-loss*.
- $takeProfitLong$: Umbral de *take-profit* porcentual estático aplicado para las operaciones del tipo alcista. Inf es no utilizar *take-profit*.
- $stopLossShort$: Umbral de s*top-loss* porcentual estático aplicado para las operaciones del tipo bajista. Inf es no utilizar *stop-loss*.
- $takeProfitShort$: Umbral de *take-profit* porcentual estático aplicado para las operaciones del tipo bajista. Inf es no utilizar *take-profit*.

Con el objetivo de evitar inconsistencias, se han sujeto restricciones a los parámetros del modelo, en particular: La restricción (5.1) evita simulaciones en que no se abren posiciones; la restricción (5.2) surge del hecho que si $ammountClosePosition$ supera $windowSize$, es equivalente a evaluar $ammountClosePosition = Inf$. Por otro lado, la restricción (5.3) se aplica a $maximumUtilizationOpen$ y $maximumUtilizationClose$, ésta se basa en el mismo principio que la restricción (5.2). Finalmente, la restricción (5.4) evita simulaciones en que no se cierren operaciones.

$$ammountOpenPosition \leq windowSize \quad \textbf{(5.1)}$$

$$ammountClosePosition \leq windowSize \mid ammountClosePosition = Inf \quad \textbf{(5.2)}$$

$$maximumUtilization \leq windowSize \mid maximumUtilization = Inf \quad \textbf{(5.3)}$$

$$TP \cong Inf \mid ammountClosePosition \cong Inf \quad \textbf{(5.4)}$$

La optimización de parámetros escapa del alcance de esta memoria, siendo apropiado maximizar retorno, minimizar *drawdown* y minimizar la duración de cada operación, obteniendo así la mayor ganancia posible, de forma rápida y con bajo riesgo. Además, para optimizar las métricas propuestas se debiese considerar otra partición para la base, evitando sobre-ajuste y pérdidas de generalidad. Por otra parte, es apropiado maximizar la ganancia condicionado a un umbral de riesgo máximo, por ejemplo, utilizar los parámetros con mayor retorno condicionado a un *drawdown* máximo de 10%.

Se consideran distintas configuraciones de parámetros con el fin de evaluar la capacidad predictiva de las recomendaciones de compraventa. Se desea determinar, entre otras cualidades, si es mejor abrir una operación cada vez que se realiza una recomendación de compraventa o esperar cierto número de recomendaciones dentro de una ventana (*clustering*). En particular se evalúan cuatro configuraciones:

1. Se considera la configuración más simplista, en ésta cada vez se encuentre una recomendación de compraventa, se realice una operación en ese sentido, en caso de encontrar una predicción en el sentido contrario, se cierre la operación y se abre en el nuevo sentido. No se considera *stop-loss* ni *take-profit*, por lo que sólo se cierran operaciones mediante la heurística. Finalmente, sólo se permite una operación abierta la vez sin *hedging*

2. En la segunda configuración también se utiliza un ancho de ventana de uno, por lo que la apertura y cierre de operaciones se realiza si al cierre de un período en particular se predice un cambio de tendencia. La única diferencia con la primera configuración radica en la utilización de umbrales de SL y TP con cierre inmediato, se considera un umbral de 3% para ambos valores en compra y venta, valor seleccionado ya que corresponde a la retracción de ZigZag con mejor desempeño.

3. En esta configuración se considera una ventana de dos períodos, la apertura de operaciones se realiza con dos recomendaciones dentro de la ventana (por lo que se abre si hay dos señales del mismo tipo consecutivas). La operación se cierra cuando aparece una señal del tipo contrario. Se mantiene el umbral de SL de 3% y se expande el de TP a

5% (simétricos para compra y venta). Se permite una operación de compra y de venta abierta como máximo a la vez con *hedging*.

4. Finalmente, se utiliza un mayor ancho de ventana (cuatro períodos), además se exigen al menos tres señales dentro de ésta para abrir una operación y una para cerrar mediante heurística. Se acota el umbral de SL a 2% y se expande el TP a 15%. Éstos se aplican al cierre de un período, es decir, sin cierre inmediato. Por otra parte se permite *hedging*, es decir, operaciones de compra y venta abiertas a la vez.

Los parámetros de las configuraciones se resumen en la Tabla 5.1 y los resultados se presentan en la Ilustración 5.3. En ésta se aprecia en cada fila las distribuciones de retorno, *drawdown* y períodos de duración (en promedio y mediana) de todas las operaciones (compra y venta) para los casos evaluados. Se muestran en rojo las operaciones cerradas mediante SL, en azul por heurísticas y en verde por TP. Con el fin de obtener simetría en los retornos (no desplazarlos hacia el lado positivo o negativo) y mantener el *drawdown* representativo, los histogramas se grafican utilizando los intervalos mostrados en la Ilustración 5.2. Finalmente, en Anexo J se presentan los resultados con detalles para operaciones de compra y venta por separado.

Parámetro	**Configuración**			
	Nº1	Nº2	Nº3	Nº4
$\boldsymbol{windowSize}$	1	1	2	4
$\boldsymbol{ammountOpenPosition}$	1	1	2	3
$\boldsymbol{ammountClosePosition}$	1	1	1	1
$\boldsymbol{maximumUtilizationOpen}$	1	1	1	1
$\boldsymbol{maximumUtilizationClose}$	1	1	1	1
$\boldsymbol{maximumPositions}$	1	1	1	1
$\boldsymbol{hedging}$	0	0	0	1
$\boldsymbol{numberPositionsToClose}$	1	1	1	1
$\boldsymbol{instantClose}$	1	1	1	0
$\boldsymbol{stopLossLong}$	Inf	3%	3%	2%
$\boldsymbol{takeProfitLong}$	Inf	3%	5%	5%
$\boldsymbol{stopLossShort}$	Inf	3%	3%	2%
$\boldsymbol{takeProfitShort}$	Inf	3%	5%	5%

Tabla 5.1: Parámetros utilizados en escenarios de simulación

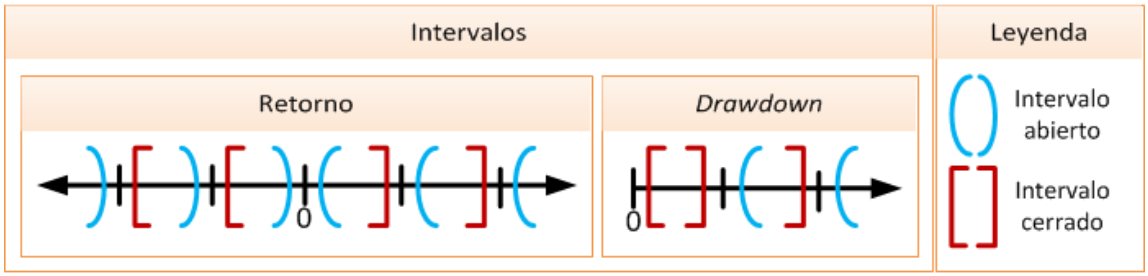

Ilustración 5.2: Intervalos utilizados en histogramas de retorno y *drawdown*

En todos los casos se aprecia que la distribución de duración de las operaciones es muy parecida en promedio y mediana (en gráficos de retorno y *drawdown*), siendo la mediana casi siempre ligeramente menor al promedio. Se infiere que no hay muchas operaciones con tiempos de duración muy largos o fuera de rango. En los gráficos de retorno se manifiesta una tendencia a que las operaciones con pérdidas duren más tiempo abiertas, por lo que una forma sencilla de mejorar los resultados es cerrando operaciones que tarden más de cierto umbral temporal. En los gráficos de *drawdown* se confirma la tendencia de mayor duración temporal con las operaciones con pérdida, además se distingue que éste tiende a decaer para mayores valores. Finalmente, cabe resaltar que con todas las configuraciones se obtienen retornos positivos tanto en promedio como en mediana, además, que esta última sea positiva revela que la mayoría de las operaciones presentan ganancias. En Anexo J se verifica que se posee ganancias tanto de compra como de venta, por lo que el sistema no obtiene retornos positivos a causa tendencia del mercado, sino debido a su potencial para anticipar cambios de tendencia.

En las configuraciones dos y tres se observa que, debido a que el umbral de SL se aplica de forma inmediata (no al precio de cierre), no se poseen operaciones con pérdidas mayores a este umbral. Por la misma razón el *drawdown* nunca supera éste valor; efecto contrario en las configuraciones uno y cuatro.

En la configuración uno se aprecia buen comportamiento en retorno, teniendo éste una forma casi Gaussiana. Se destaca un leve desplazamiento a retornos positivos con un promedio de 0.5% y mediana de 1% (por operación). Se advierte que la duración de las operaciones con retornos positivos ronda por los cuatro a seis períodos en promedio y mediana, por lo tanto, en general la mayoría de las operaciones no está abierta por más de seis días. Además, la clara mayor duración de las operaciones con perdida resaltan el potencial de utilizar un umbral de tiempo máximo. Se observa una transacción con pérdidas sobre 40%, seguida por la segunda mayor alrededor de 15, por lo que la utilización de un SL de seguridad, aunque sea amplio, siempre es recomendada.

La segunda configuración presenta un comportamiento muy interesante, resaltando que la mayoría de las transacciones son cerradas mediante SL o TP, habiendo muy pocas cerradas mediante heurísticas con retornos en el rango $[-2\%, 2\%]$. Se destaca que debido a la utilización de SL, el *drawdown* promedio es bastante menor al caso anterior, no sucediendo lo mismo con la mediana. Por otro lado, el retorno promedio es casi la mitad y la mediana relativamente menor. Lo que podría ser indicio de que este caso presenta menores retornos que el caso anterior, pero a menor riesgo. Finalmente, cabe notar que la utilización de SL es adecuada, esto se infiere ya que en el gráfico de *drawdown* se aprecian operaciones en verde en el rango 2% a 3%, por lo que hubo operaciones que empezaron con pérdida y terminaron con ganancias, dando rango suficiente para que el precio oscile antes del cambio de tendencia anticipado.

En la tercera configuración se aprecia menor número de operaciones que en las configuraciones anteriores, esto se adjudica a que la heurística de apertura de operaciones exige dos recomendaciones del mismo tipo consecutivas. No se puede determinar si la utilización de clústeres y expansión del rango de TP mejoran los resultados, ya que, en promedio se obtienen mayores retornos, pero menores en mediana. Por otra parte, el promedio y mediana de *drawdown* prácticamente no varía.

En la última configuración, se advierte menor retorno promedio y mediano que con las tres anteriores, además de un pequeño aumento de *drawdown*, el que se adjudica a la utilización de los umbrales de SL y TP al precio de cierre y no de forma inmediata. Se aprecia que algunas operaciones que topan el umbral de TP tienden a seguir la dirección de la transacción, por lo que podría ser apropiado aplicar cierre inmediato al SL para acotar pérdidas y no al TP con el fin de no limitar posibles retornos mayores.

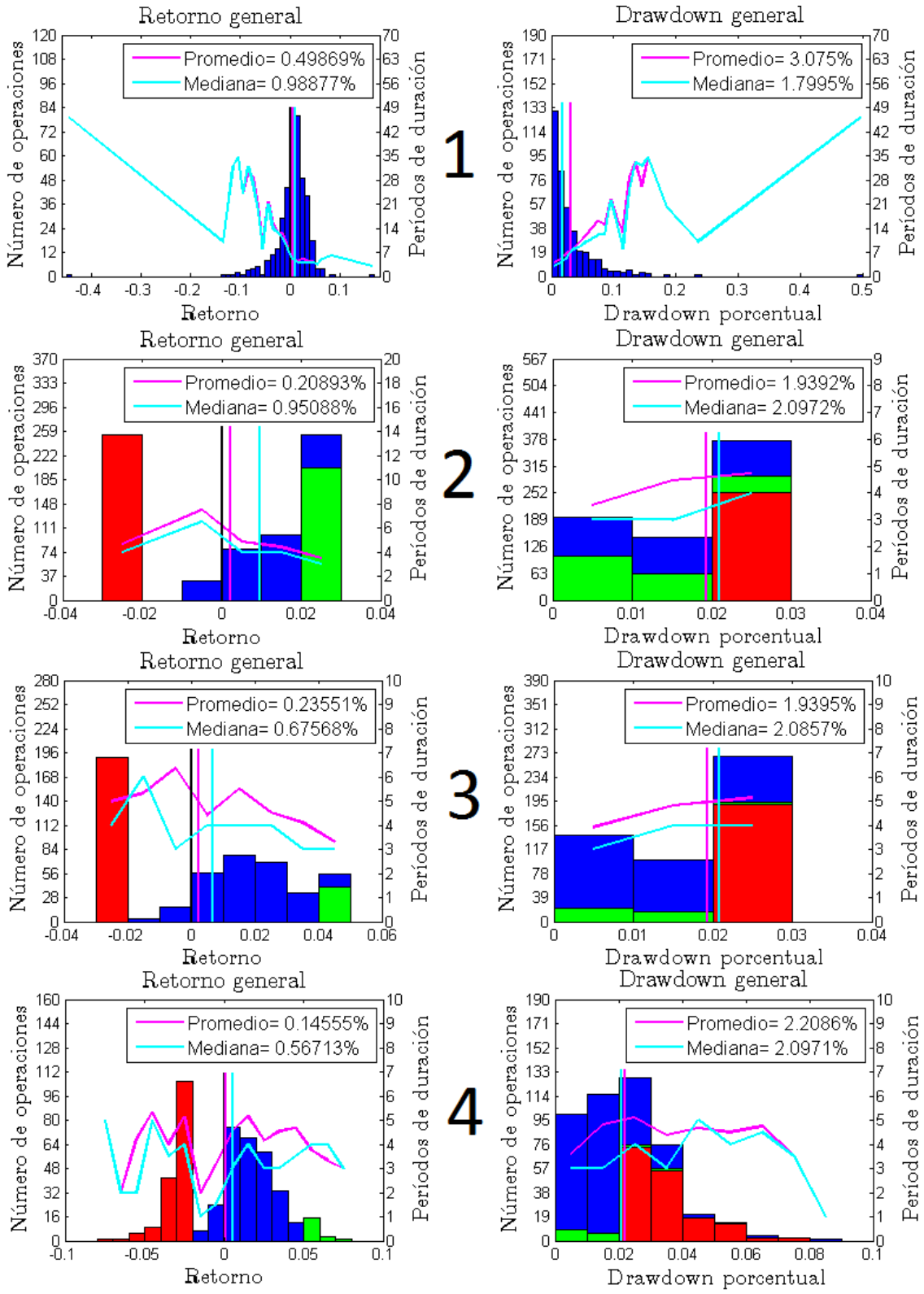

Ilustración 5.3: Retorno y *drawdown* generales para configuraciones simuladas

Configuración	Retorno [%]		*Drawdown* [%]	
	Promedio	Mediana	Promedio	Mediana
1	0.62333	0.88719	0.2803	1.9608
2	0.187	0.70569	1.9787	2.1268
3	0.20065	0.5787	1.9702	2.103
4	0.1867	0.46175	2.1829	2.1144

Tabla 5.2: Métricas por operación para compra con configuraciones simuladas

Configuración	Retorno [%]		*Drawdown* [%]	
	Promedio	Mediana	Promedio	Mediana
1	0.37955	1.0556	3.335	1.7144
2	0.23316	1.1245	1.8884	2.0248
3	0.26837	1.1111	1.9106	2.0192
4	0.10666	0.68524	2.2329	2.0971

Tabla 5.3: Métricas por operación para venta con configuraciones simuladas

Configuración	Retorno [%]		*Drawdown* [%]	
	Promedio	Mediana	Promedio	Mediana
1	0.49869	0.98877	3.075	1.7995
2	0.21068	0.94884	1.9323	2.0811
3	0.23551	0.67568	1.9395	2.0857
4	0.14555	0.56713	2.2086	2.0971

Tabla 5.4: Métricas por operación en general con configuraciones simuladas

Se desea determinar cuál de los casos evaluados presenta mayores retornos. Para esto, se genera un portafolio equidistribuido compuesto por las 40 acciones en estudio, es decir, se asigna la misma inversión sobre todas las acciones. La evaluación se realiza sobre el período de testeo y no se consideran comisiones.

En la Ilustración 5.4 se muestra la curva de capital temporal para las cuatro configuraciones evaluadas. Se destaca que la configuración que presenta mayor retorno es la más simple, con un crecimiento relativamente suave (sin mucha varianza), a excepción de una gran caída alrededor del día 45, período en el cual las otras configuraciones presentan ganancias, por lo que podría ser apropiado evaluar una combinación de configuraciones con el fin de disminuir el riesgo. Se aprecia que todos los casos expuestos presentan retornos positivos al final del período de testeo, además, la configuración uno nunca posee menos capital que al principio, aumentando éste en un total de 5.08%.

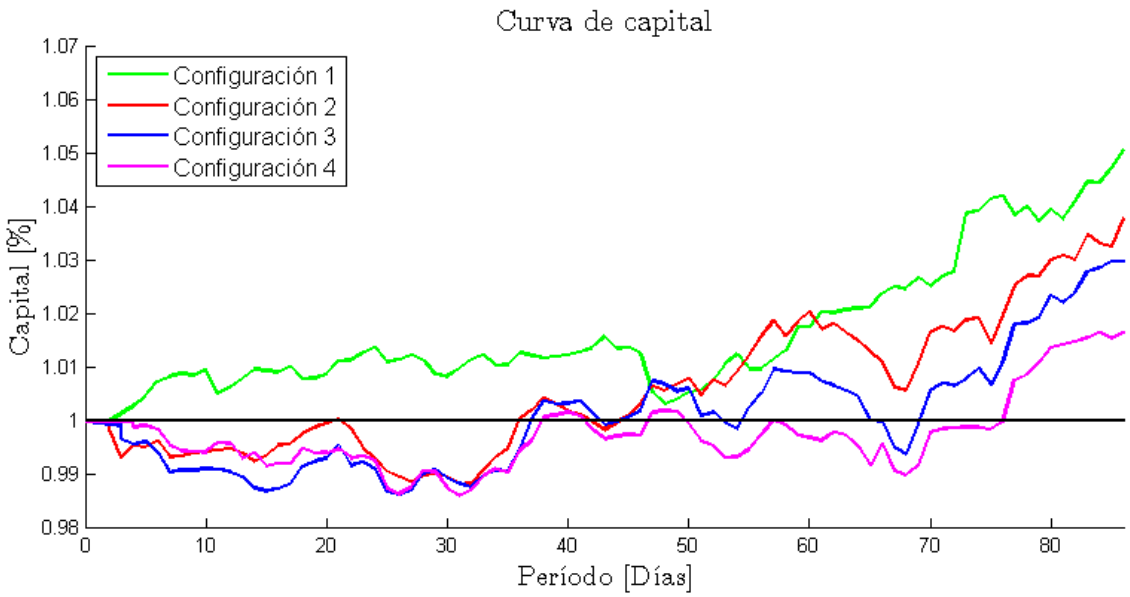

Ilustración 5.4: Curva de capital en período de testeo para cuatro escenarios simulados

En la Ilustración 5.5 se presentan las transacciones realizadas en CORPBANCA durante el período de testeo, en ésta se utilizan los parámetros del simulador de inversiones con la configuración uno. Las transacciones se presentan con líneas verdes o rojas dependiendo si el retorno fue positivo o negativo respectivamente, además se une la fecha y precio de apertura con la fecha y precio cierre de cada operación. Se aprecia que durante los cuatro meses (aprox.) se realizaron 18 transacciones, trece con retornos positivos y cinco con negativos, obteniendo un *Accuracy* de 72.2%. Si la inversión hubiera sido realizada sólo en este activo, y no repartido el capital entre las 40 acciones, se hubiese obtenido un retorno de 29.87% (sin considerar comisiones). Por lo que la optimización, tanto de parámetros como de portafolio, presenta gran atractivo para maximizar el retorno que la metodología propuesta puede obtener del mercado.

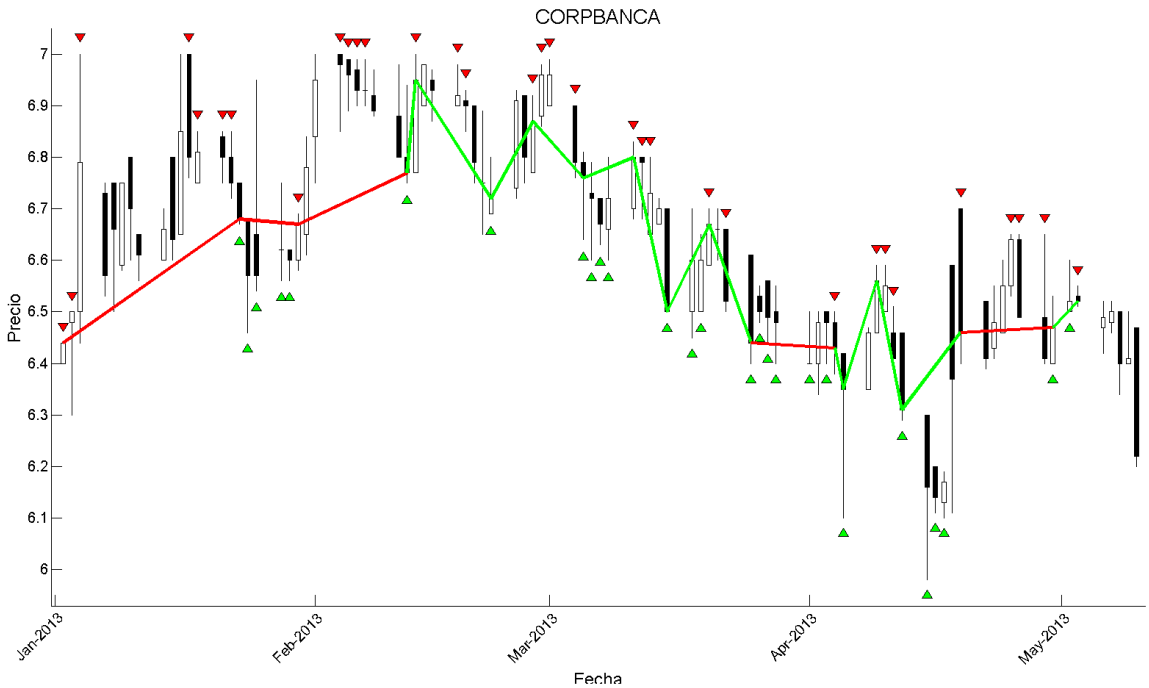

Ilustración 5.5: Transacciones de caso uno sobre período de testeo en COPRBANCA

Capítulo 6

Conclusiones

En esta memoria se ha investigado la existencia de patrones que permitan anticipar cambios de tendencia de las acciones que conforman el IPSA 2013. Para esto, se ha desarrollado una metodología que se basa en la extracción de atributos a partir de indicadores técnicos y de datos transaccionales de Insiders, etiquetado de mínimos y máximos locales mediante retracciones de zigzag, mapeo mediante herramientas de aprendizaje de máquina, análisis de poder predictivo de los atributos extraídos y evaluación mediante simulación del sistema en general.

Respecto a la selección de características, se concluye que la utilización de FS y BE son muy útiles para analizar los atributos del modelo, pues seleccionan los con mayor valor predictivo y los con mayor ruido respectivamente, además de aumentar el rendimiento sobre la métrica utilizada y acelerar el entrenamiento de los modelos de aprendizaje supervisado (debido a la selección de un subconjunto atributos).

En cuanto a la utilización de AUC como métrica, se concluye que ésta facilita la evaluación de los modelos en todo tipo de bases, ya que al no utilizar un umbral como lo hace *Accuracy*, ésta representa mejor el comportamiento de predicción del modelo sobre los datos. Además, se destaca la utilidad de OHM para la selección del mejor modelo, ya que integra las propiedades de las métricas de AUC, *Accuracy*, *F1-Score* y PD.

A modo de conclusión de los modelos predictivos utilizados, se establece que utilizar regresión logística es apropiado para la selección de atributos, ya que posee gran capacidad predictiva y no necesita probar distintas configuraciones de parámetros, acelerando así el costoso proceso con excelentes resultados. Por otra parte, en cuanto a la extracción de patrones altamente no lineales, los modelos BPNN, SVM y SBM entregan resultados muy satisfactorios, obteniendo resultados con *Accuracy* por sobre 70%. Se determina que la utilización de combinaciones de predicciones sobre la base no balanceada es una buena metodología, pues se aprecia que los modelos combinados poseen mejores características (en base a las métricas) que utilizando las predicciones por si solas.

Los resultados predictivos indican que las variables extraídas presentan gran valor anticipativo de cambios de tendencia del mercado. Además, la utilización del indicador ZigZag con la posterior extensión no causal de variable dependiente entrega suficientes etiquetas de compraventa, haciendo posible la utilización de algoritmos de aprendizaje supervisado. Sin embargo, al ser un primer acercamiento y poseer parámetros ajustables, puede ser depurado.

Se concluye que la utilización de múltiples cambios porcentuales de retracción de precio (mediante indicador ZigZag), es una metodología adecuada, ya que es equivalente a extraer distintos ciclos del mercado, permitiendo así la evaluación y utilización del ciclo al que mejor se adapten los atributos extraídos.

De los resultados obtenidos en el simulador de inversiones, se concluye que el sistema, como primera implementación, posee gran potencial, pues logra extraer retornos positivos tanto en promedio como en mediana, con relativamente bajo *drawdown* para operaciones de compra y venta por separado, es decir, los resultados no se atribuyen a tendencia del mercado.

En cuanto al objetivo de encontrar patrones con valor anticipativo de cambio de tendencia, los resultados indican que la hipótesis se válida, ya que se logran predicciones con AUC sobre 80% en la base de validación y retornos positivos en la posterior simulación de inversiones en la base de testeo, lo que se puede atribuir a que el mercado en estudio es poco profundo, ilíquido o ineficiente.

En general, se concluye que la metodología propuesta representa un buen comienzo para extraer valor adicional del mercado financiero. Sin embargo, el trabajo realizado es el comienzo de lo que podría ser un sistema más complejo, con mayores retornos y menor riesgo. La facilidad que entrega la metodología propuesta es que puede ser seccionada y optimizada por bloques (Anexo K), optimizando así el rendimiento general del sistema.

Finalmente, se destaca el cumplimiento a cabalidad de los objetivos propuestos, destacando que los indicadores utilizados poseen poder anticipativo de cambios de tendencia del mercado. El indicador ZigZag permite la adecuada aplicación de minería de datos, se evalúa el poder predictivo de distintos tipos de atributos y, finalmente, se evalúa cuantitativamente la metodología propuesta mediante un simulador de inversiones.

6.1. Trabajo futuro

Como trabajo futuro se propone:

- Agregar más indicadores con capacidad predictiva de cambios de tendencia. En particular, se propone agregar variables de divergencia del indicador de Acumulación/Distribución de William; variables de cruces y divergencia del Índice Direccional Promedio (*Average Directional Index* <ADX>). Además, se propone extender el período de activación de cada atributo bajo la premisa de que un atributo puede tener validez por más períodos que sólo en el que fue activado (Ej: una divergencia alcista regular puede ser válida por diez períodos).

- Se propone la selección de características que genere mejores predicciones que la utilización de FS o BE, en particular, como exploran Yang y Honavar [47], se pueden utilizar algoritmos genéticos en la selección de éstos.

- En cuanto a la fase de modelamiento sobre la base no balanceada, se propone la combinación de modelos utilizando una red Bayesiana que considere las dependencias entre variables [48] y una metodología de programación genética (basada en Algoritmos Evolutivos), la que genera árboles de reglas condicionales (en base a salidas de modelos) para adaptarse a la función objetivo.

- Respecto al simulador de inversiones se puede:

    - Optimizar los parámetros mediante heurísticas o algoritmos evolutivos.
    - Aumentar la resolución de las variables utilizadas.
    - Implementar SL y TP dinámicos. Por ejemplo, ajustar el SL cuando el precio avance a favor de acuerdo a las oscilaciones (cruces por umbrales de compra/venta) del SO.
    - Utilizar distintos umbrales de “cubrir gasto” (*Break Even* <BE>), umbral en el cual el SL se mueve a cero (o con ganancias marginales). En caso que el activo retorne no se tienen pérdidas y/o se aseguran pequeñas ganancias.

Glosario

AUC *Area Under Curve*
BE *Backward Elimination*
BPNN *Backpropagation Neural Network*
EMA *Exponential Moving Average*
FIR *Finite Impulse Response*
FPR *False Positive Rate*
FS *Forward Selection*
IIR *Infinite Impulse Response*
IPSA Índice de Precios Selectivos de Acciones
KDD *Knowledge Discovery from Databases*
MA *Moving Average*
MACD *Moving Average Convergence/Divergence*
MLP *Multi-Layer Perceptron*
NB *Naive Bayes*
OHLC *Open, High, Low, Close*
OHM *Overall Harmonic Mean*
ROC *Receiver Operating Characteristic*
RSI *Relative Strength Index*
SBM *Similarity Based Method*
SKA *Space Kernel Analysis*
SL *Stop-loss*
SMA *Simple Moving Average*
SO *Stochastic Oscillator*
SVM *Support Vector Machine*
TP *Take-profit*
TPR *True Positive Rate*
TS *Trading Simulator*

Bibliografía

[1] FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. 1996. The KDD process for extracting useful knowledge from volumes of data.Communications of the ACM, vol. 39, no 11, p. 27-34.

[2] LO, Andrew W.; MACKINLAY, Archie Craig. 1998. Stock market prices do not follow random walks: Evidence from a simple specification test. Review of financial studies, vol. 1, no. 1, pp. 41-66.

[3] ODOM, Marcus D.; SHARDA, Ramesh. 1990. A neural network model for bankruptcy prediction. En: Neural Networks. 1990 IJCNN International Joint Conference on. IEEE. pp. 163-168.

[4] MIN, Jae H.; LEE, Young-Chan. 2005. Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. Expert systems with applications, vol. 28, no 4, pp. 603-614.

[5] DEL BRIO, Esther B.; MIGUEL, Alberto; PEROTE, Javier. 2002. An investigation of insider trading profits in the Spanish stock market. The Quarterly Review of Economics and Finance, vol. 42, no 1, pp. 73-94.

[6] BHATTACHARYA, Utpal; DAOUK, Hazem. 2002. The world price of insider trading.The Journal of Finance. vol. 57, no 1, pp. 75-108.

[7] FOIX, Cristian; WEBER, Richard. 2007. Pronóstico del precio del cobre mediante redes neuronales. INGENIERIA DE SISTEMAS, pp. 63.

[8] SCHWARTZ, Eduardo S. 1997. The stochastic behavior of commodity prices: Implications for valuation and hedging. The Journal of Finance, vol. 52, no 3, p. 923-973.

[9] YANG, Jian; BESSLER, David A.; LEATHAM, David J. 2001. Asset storability and price discovery in commodity futures markets: a new look. Journal of futures markets, vol. 21, no 3, pp. 279-300.

[10] DONOHO, Steve. 2004. Early detection of insider trading in option markets. En: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. pp. 420-429.

[11] JOHANSEN, Anders; SORNETTE, Didier. 2002. Large stock market price drawdowns are outliers. Journal of Risk, vol. 4, p. 69-110.

[12] CHEKHLOV, Alexei; URYASEV, Stanislav; ZABARANKIN, Michael. 2005. Drawdown measure in portfolio optimization. International Journal of Theoretical and Applied Finance, vol. 8, no 01, pp. 13-58.

[13] ACAR, Emmanuel; TOFFEL, Robert. 2000. Stop-loss and Investment Returns. Investment Conference, faculty an Institute of Actuaties, Hatfield Heath.

[14] CASTAÑEDA, Juan E.; HERNÁNDEZ, Mauro. 2011. Burbujas especulativas en los precios de los activos (oficios, ferrocarriles, inmuebles y bolsa). En: X Congreso Internacional de la AEHE. Carmona (Sevilla), España. Universidad Pablo de Olavide, pp. 26.

[15] NEELY, Christopher J. 1997. Technical analysis in the foreign exchange market: a layman's guide. Federal Reserve Bank of St. Louis Review, vol. 79, no September/October 1997.

[16] MURPHY, John J. 1999. Technical Analysis of the Financial Markets, A Comprehensive Guide to Trading Methods and Applications. New York Institute of Finance. pp. 1-32.

[17] DZIKEVICIUS, A.; SARANDA, S.; KRAVCIONOK, A. 2010. The accuracy of simple trading rules in stock markets. Economics & Management.

[18] KIRKPATRICK II, C. D.; DAHLQUIST, J. 2010. Technical analysis: the complete resource for financial market technicians. FT press. 2a ed. pp. 275-297.

[19] BOISSARD, Johan.2012. Applications and Uses of Digital Filters in Finance. Master of Science in Management, Technology and Economics. Swiss, Zurich, Department of Management, Technology and Economics, Swiss Federal Institute of Technology.

[20] APPEL, Gerald. 2005. Technical analysis: power tools for active investors. FT Press.

[21] HARTLE, Thom. 1991. Moving Average Convergence/Divergence (MACD). Stocks & Commodities vol. 9, no. 3 pp. 104.

[22] VAKKUR, Mark. 1997. The Moving Average Convergence/Divergence. Stocks & Commodities vol. 15, no. 4 pp. 145-153.

[23] LEBARON, Blake. 2000. The stability of moving average technical trading rules on the Dow Jones Index. Deriv. Use Trad. Regul, vol. 5, p. 324-338.

[24] WILDER, J. Welles. 1978. New concepts in technical trading systems. NC: Trend Research. McLeansville. pp. 70.

[25] ALFARO, Rodrigo; SAGNER, Andres. 2010. Financial Forecast for the Relative Strength Index.

[26] ALARCÓN, Felipe; PINCHEIRA, Pablo; SELAIVE, Jorge. 2007. Tipo de cambio nominal chileno: predicción basada en análisis técnico. Economía Chilena.

[27] Morris, G. 2006. Candlestick Charting Explained, Chapter 8. Candle Pattern Performance. McGraw Hill Professional.

[28] LUISI, J. 1997. The Stochastic Oscillator. Technical analysis of stocks and commodities-magazine, ed. 15, pp. 77-78.

[29] Stochastic Oscillator. http://stockcharts.com/help/doku.php?id=chart_school:technical_indicators:_stochastic_oscillato [consulta: 20 Noviembre 2013]

[30] MARTIN J. PRING. 2002. En: Technical Analysis Explained. 4ª ed. New-York. McGraw-Hill. pp. 211-236.

[31] CASTI, J.; MEYER, J.; TAYLOR, R. P. 2011. Social Mood, "Deep" History, and the Elliott Wave Principle.

[32] AIZERMAN, A.; BRAVERMAN, Emmanuel M.; ROZONER, L. I. 1964. Theoretical foundations of the potential function method in pattern recognition learning. Automation and remote control, vol. 25, pp. 821-837.

[33] CORTES, Corinna; VAPNIK, Vladimir. 1995. Support-vector networks. Machine learning, vol. 20, no 3, p. 273-297.

[34] HSU, Chih-Wei, et al. 2003. A practical guide to support vector classification.

[35] GONG, L.; SCHONFELD, D. 2009. Space Kernel Analysis, Acoustics, Speech and Signal Processing. ICASSP 2009. Taipei. IEEE International Conference. pp.1577-1580.

[36] WEGERICH, S.; XU, X. 2003. A performance comparison of similarity-based and kernel modeling techniques. Knoxville, TN. In Proceedings of MARCON 2003.

[37] WEGERICH, Stephan W. 2004. Similarity based modeling of time synchronous averaged vibration signals for machinery health monitoring. En Aerospace Conference, 2004. Proceedings. 2004 IEEE. IEEE. pp. 3654-3662.

[38] HOLTAN, T.; WHEELER, T. 2003. Using real-time predictive condition monitoring to increase coal plant asset availability. Coal-Gen.

[39] NIEMAN, W.; OLSON, R. 2001. Early detection of signal or process variation in the co-generation plant at U.S. Steel, Gary Works. Proceedings of Turbo Expo: Land, Sea, Air.

[40] WEGERICH, S.; WILKS, A.; PIPKE, R. 2003. Nonparametric modeling of vibration signal features for equipment health monitoring. En Proceedings of the IEEE Aerospace Conference. pp. 3113-3121.

[41] LABATUT, Vincent; CHERIFI, Hocine. 2012. Accuracy measures for the comparison of classifiers.

[42] FAWCETT, Tom. 2004. ROC graphs: Notes and practical considerations for researchers. Machine learning, vol. 31, pp. 1-38.

[43] BRADLEY, Andrew P. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern recognition, vol. 30, no 7, pp. 1145-1159.

[44] FAWCETT, Tom. 2006. An introduction to ROC analysis. Pattern recognition letters, vol. 27, no 8, pp. 861-874.

[45] Consorcio. http://www.ccbolsa.cl [consulta: 02 mayo 2013]

[46] Insiders. http://www.insiders.cl/ [consulta: 05 mayo 2013]

[47] YANG, Jihoon; HONAVAR, Vasant. 1998. Feature subset selection using a genetic algorithm. En Feature extraction, construction and selection. Springer US. pp. 117-136.

[48] ZHANG, Harry. 2004. The optimality of naive Bayes. A A, vol. 1, no 2, pp. 3.

Anexos

Anexo A: Acciones que componen el IPSA 2013

N°	Ticker	Empresa	Ponderación
1	AESGENER	AESGENER S.A.	2,07
2	AGUAS-A	AGUAS ANDINAS S.A., SERIE A	2,82
3	ANDINA-B	EMBOTELLADORA ANDINA S.A. SERIE B	1,97
4	ANTARCHILE	ANTARCHILE S.A.	2,17
5	BCI	BANCO DE CREDITO E INVERSIONES	3,5
6	BESALCO	BESALCO S.A.	0,49
7	BSANTANDER	BANCO SANTANDER-CHILE	3,81
8	CAP	CAP S.A.	3,3
9	CCU	COMPANIA CERVECERIAS UNIDAS S.A.	1,88
10	CENCOSUD	CENCOSUD S.A.	6,46
11	CFR	CFR PHARMACEUTICALS S.A.	0,63
12	CHILE	BANCO DE CHILE	3,72
13	CMPC	EMPRESAS CMPC S.A.	4,18
14	COLBUN	COLBUN S.A.	3,2
15	CONCHATORO	VINA CONCHA Y TORO S.A.	1,14
16	COPEC	EMPRESAS COPEC S.A.	8,66
17	CORPBANCA	CORPBANCA	0,89
18	CRUZBLANCA	CRUZ BLANCA SALUD S.A.	0,52
19	ECL	E.CL S.A.	1,2
20	EMBONOR-B	COCA-COLA EMBONOR S.A., SERIE B	0,51
21	ENDESA	EMPRESA NACIONAL DE ELECTRICIDAD S.A.	6,79
22	ENERSIS	ENERSIS S.A.	5,8
23	ENTEL	EMP. NACIONAL DE TELECOMUNICACIONES S.A.	2,56
24	FALABELLA	S.A.C.I. FALABELLA	6,38
25	FORUS	FORUS S.A.	0,73
26	HITES	EMPRESAS HITES S.A.	0,15
27	IAM	INVERSIONES AGUAS METROPOLITANAS S.A.	1,13
28	ILC	INVERSIONES LA CONSTRUCCION S.A.	0,79
29	LAN	LATAM AIRLINES GROUP S.A.	8,02
30	NUEVAPOLAR	EMPRESAS LA POLAR S.A.	0,33
31	PARAUCO	PARQUE ARAUCO S.A.	1,54
32	PAZ	PAZ CORP S.A.	0,13
33	RIPLEY	RIPLEY CORP S.A.	1,12
34	SALFACORP	SALFACORP S.A.	0,53
35	SK	SIGDO KOPPERS S.A.	0,73
36	SM-CHILE B	SOCIEDAD MATRIZ BANCO DE CHILE, SERIE B	2,22
37	SMSAAM	SOCIEDAD MATRIZ SAAM S.A.	0,85
38	SONDA	SONDA S.A.	1,5
39	SQM-B	SOC. QUIMICA MINERA DE CHILE S.A., SERIE B	5,05
40	VAPORES	COMPANIA SUDAMERICANA DE VAPORES S.A.	0,55

Tabla A.1: Acciones que componen el IPSA 2013

Anexo B: Pseudocódigo para cálculo de ZigZag

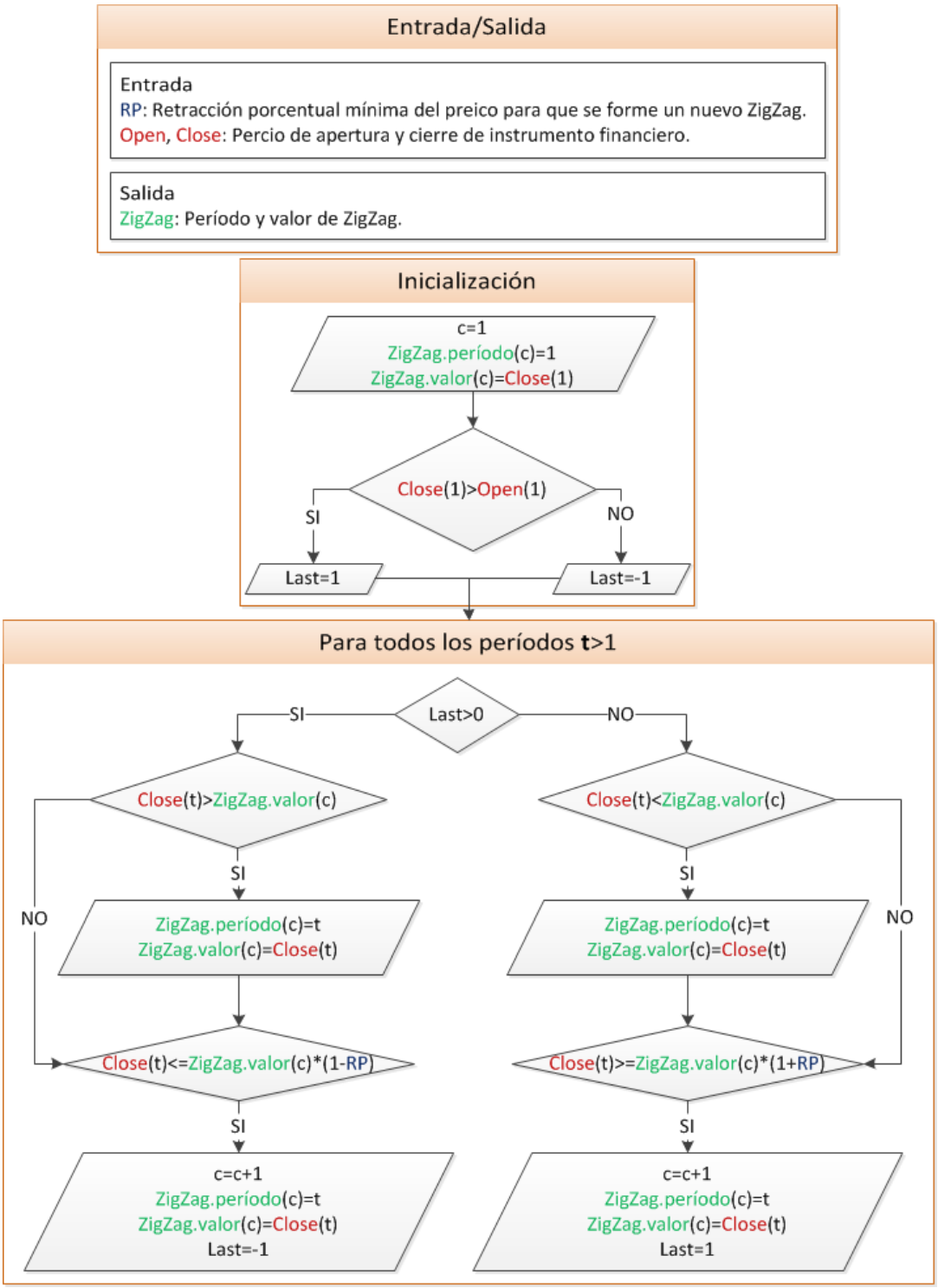

Ilustración B.1: Pseudocódigo en diagrama de bloques para Cálculo de ZigZag

Anexo C: Pseudocódigo para obtención de divergencias

Entrada/Salida

Entrada
Base: Estructura con precios OHLC de instrumento financiero.
Par: Estructura con parámetros.
Indicators: Indicador principal y filtro secundario (opcional).

Salida
Divergence: Estructura con divergencias.

Par

Par.lastRegularPeaks: Número de peaks a considerar en cálculo de divergencia regular.
Par.lastHiddenPeaks: Número de peaks a considerar en cálculo de divergencia oculta.
Par.lastPeriods: Máximo número de períodos a considerar en el pasado para cálculo de divergencias.
Par.searchNeighborhood: Número de períodos a considerar en vecindad (para pasado y futuro).
Par.maximumPeriodDiference: Relación de períodos máxima a considerar en divergencias triples.
Par.overboughtThreshold: Umbral mínimo de primer peak para considerar divergencia.
Par.offset: Sesgo de indicadores en "*overboughtThreshold*" para divergencias alcistas.

Inicialización

```
//Calcular divergencias bajistas con precio máximo
Divergence.short.simple = getDivergence(Base.High, Par, mainIndicator, secondIndicator)
//Calcular divergencias alcistas con precio mínimo e indicador invertido
Divergence.long.simple = getDivergence(Base.Low, Par, -mainIndicator+offset, -secondIndicator+offset)
//Calcular divergencias triples a partir de las simples
Divergence = getTripleDivergence(Divergence, Par)
```

getDivergences

```
Divergence = getDIvergences(highPrice, Par, mainIndicator, secondIndicator)
//Obtener todos los peaks a priori aumenta la eficiencia de cálculo
Peaks = getIndicatorExtremes(mainIndicator, secondIndicator)
//Sólo considerar peaks que sobrepasen umbral de sobrecompra
peaksAbove = peak(mainIndicator(peak)>=Par.overboughtThreshold)
for t=3:periods-1 //Recorrer los períodos
        //Verificar formación de divergencia parcial, la señal se aplica al período siguiente
        //El criterio de div. parcial es esperar dos períodos en alza y uno en baja como confirmación
        if mainIndicator(t)>=mainIndicator(t-1) && mainIndicator(t-1)>=mainIndicator(t-2) &&
          mainIndicator(t)>=mainIndicator(t+1)
                //Obtener peaks entre Par.lastPeriods y ahora
                peaksAux = peaksAbove(peaksAbove>=t-Par.lastPeriods & peaksAbove<t-1)
                //Obtener estructura de divergencia regular
                Divergence = getRegularDivergence(t,Price,peaksAux,Par, Divergence, mainIndicator)
                //Obtener estructura de divergencia oculta
                Divergence = getHiddenDivergence(t,Price,peaksAux,Par, Divergence, mainIndicator)
        end
end
```

Ilustración C.1: Pseudocódigo para extraer divergencias (inicialización <parte 1/5>)

getRegularDivergence

```
Divergence = getCLassicDivergence(actualPeriod, highPrice, peaksAux, Par, Divergence, mainIndicator)
//Obtener precio máximo en vecindad; por causalidad hacia futuro sólo se considera un período
[actualHigherPrice,actualHigherPeriod]=max(highPrice(max(1,actualPeriod-
Par.searchNeighborhood):actualPeriod+1))
//Obtener último precio mayor que el actual como límite de búsqueda de divergencias
lastHigherPeriod=find(highPrice(1:actualHigherPeriod-1)>actualHigherPrice,"last")
// Si no existe un percio mayor, se busca desde el primer período
If isempty(lastHigherPeriod)
        lastHigherPeriod=1
end
peaksRegular=find(peaksAux>=lastHigherPeriod) //Buscar peaks después de "lastHigherPeriod"
If ~isempty(peaksRegular) // Si existen peaks después de "lastHigherPeriod"
        //Sólo considerar los últimos "Par.lastClassicPeaks " peaks
        peaksRegular=peaksRegular(max(1,length(peaksRegular)-Par.lastRegularPeaks+1):end)
        maxPeak=-Inf // Inicializar peak máximo como menos infinito
        pr=length(peaksRegular) //Número de peaks en "peaksRegular"
        for j=1:pr  //Recorrer todos los peaks en búsqueda de divergencias
                //Ir hacia el pasado, si peak es menor que el peak máximo actual, se omite
                if mainIndicator(peaksRegular(pr-j+1))<maxPeak
                        continue  //Ir a siguiente ciclo
                else
                        maxPeak=mainIndicator(peaksRegular(pr-j+1)) //Actualizar valor máximo de peak
                end
                //Obtener precio máximo dentro de vecindad, a futuro sólo se considera un período por causalidad
                [higherPeakPrice,higherPeakPeriod]=max(max(1,peaksRegular(pr-j+1)-Par.searchNeighborhood:
                min(peaksRegular(pr-j+1)+Par.searchNeighborhood,kength(highPrice),actualPeriod+1)))
                //Verificar divergencia regular bajista
                if mainIndicator(actualperiod)<mainIndicator(peaksRegular(pr-j+1)) &&
                  actualHigherPrice>higherPeakPrice
                        //Verificar colisión de divergencia con "highPrice" y "mainIndicator"
                        if ~checkCollision(highPrice(higherPeakPeriod:actualHigherPeriod)) &&
                           ~checkCollision(mainIndicator(peaksRegular(pr-j+1):actualPeriod))
                                //Verificar si peak es parcial o confirmado (si el período pertenece a "peaks")
                                if find(actualPeriod,peaksAux) //Agregar divergencia como confirmada
                                        Divergence.confirmed.regular.price.first(end+1)=higherPeakPeriod
                                        Divergence.confirmed.regular.price.last(end+1)=actualHigherPeriod
                                        Divergence.confirmed.regular.indicator.first(end+1)=peaksRegular(pr-j+1)
                                        Divergence.confirmed.regular.indicator.last(end+1)=actualPeriod
                                else  //Agregar divergencia como parcial
                                        Divergence.partial.regular.price.first(end+1)=higherPeakPeriod
                                        Divergence.partial.regular.price.last(end+1)=actualHigherPeriod
                                        Divergence.partial.regular.indicator.first(end+1)=peaksRegular(pr-j+1)
                                        Divergence.partial.regular.indicator.last(end+1)=actualPeriod
                                end
                        end
                end
        end
end
```

Ilustración C.2: Pseudocódigo para extraer divergencias (divergencias regulares <parte 2/5>)

getHiddenDivergence

```
Divergence = getHiddenDivergence(actualPeriod, highPrice, peaksAux, Par, Divergence, mainIndicator)
//Obtener precio máximo en vecindad; por causalidad hacia futuro sólo se considera un período
[actualHigherPrice,actualHigherPeriod]=max(highPrice(max(1,actualPeriod-Par.searchNeighborhood):actualPeriod+1))
//Obtener último peak de "mainIndicator" como límite de búsqueda de divergencias ocultas
lastHigherPeak=find(mainIndicator(peaksAux)>mainIndicator(actualPeriod),"last")
//Si no hay peaks mayores, buscar desde el primer período
If isempty(lastHigherPeak)
        lastHigherPeak=1
end
peaksHidden=peaksAux(peaksAux>=lastHigherPeak) //Buscar peaks después de "lastHigherPeak"
If ~isempty(peaksHidden) //Si hay peaks después de "lastHigherPeak"
        //Sólo considerar los últimos "Par.lastHiddenPeaks " peaks
        peaksHidden=peaksHidden(max(1,length(peaksHidden)-Par.lastHiddenPeaks+1):end)
        maxPrice=-Inf //Inicializar precio máximo como menos infinito
        ph=length(peaksHidden) //Número de peaks en "peaksHidden"
        for j=1:ph  //Recorrer todos los peaks en reversa
                //Obtener precio máximo en vecindad, a futuro sólo se considera un período por causalidad
                [higherPeakPrice,higherPeakPeriod]=max(highPrice(max(1,peaksHidden(ph-j+1)-Par.searchNeighborhood):
                min(peaksHidden(ph-j+1)-Par.searchNeighborhood),length(highPrice),actualPeriod+1)))
                // Si precio es menor que máximo actual, se omite
                if higherPeakPrice<maxPrice
                        continue  //Ir a siguiente ciclo
                else
                        maxPrice=higherPeakPrice //Actualizar precio máximo
                end
                //Verificar divergencia oculta bajista
                if mainIndicator(actualperiod)>mainIndicator(peaksHidden(ph-j+1)) &&
                  actualHigherPrice<higherPeakPrice
                        //Verificar colisión de divergencia con with "highPrice " y "mainIndicator"
                        if ~checkCollision(highPrice(higherPeakPeriod:actualHigherPeriod)) &&
                          ~checkCollision(mainIndicator(peaksHidden(ph-j+1):actualPeriod))
                                //Verificar si peak es parcial o confirmado (si el período pertenece a "peaks")
                                if find(actualPeriod,peaksAux) //Agregar divergencia como confirmada
                                        Divergence.confirmed.hidden.price.first(end+1)=higherPeakPeriod
                                        Divergence.confirmed.hidden.price.last(end+1)=actualHigherPeriod
                                        Divergence.confirmed.hidden.indicator.first(end+1)=peaksHidden(ph-j+1)
                                        Divergence.confirmed.hidden.indicator.last(end+1)=actualPeriod
                                else  //Agregar divergencia como parcial
                                        Divergence.partial.hidden.price.first(end+1)=higherPeakPeriod
                                        Divergence.partial.hidden.price.last(end+1)=actualHigherPeriod
                                        Divergence.partial.hidden.indicator.first(end+1)=peaksHidden(ph-j+1)
                                        Divergence.partial.hidden.indicator.last(end+1)=actualPeriod
                                end
                        end
                end
        end
end
```

Ilustración C.3: Pseudocódigo para extraer divergencias (divergencias ocultas <parte 3/5>)

checkCollision

```
collisionDetected =checkCollision(valueVector)
collisionDetected = 0 //Inicializar sin colisión detectada
valueRange=valueVector(end)-valueVector(1)
valueSlope=valueRange(length(valueVector)-1) //Calcular pendiente
for t=3:length(valueVector) -2 //Iterar valores intermedios
        if valueVector(t)>(valueVector(1)+valueSlope*(t-1)) //Verificar colisión de recta
                collisionDetected = 1
                return //Finalizar iteración
        end
end
```

getIndicatorExtremes

```
peaksPosition = getIndicatorExtremes(mainIndicator, secondIndicator)
//Buscar peaks en "secondIndicator " y luego en "mainIndicator"
//Sólo considerar como peak: Primer peak en "mainIndicator" antes de peak en "secondIndicator"
while (i=3 , i<=length(secondIndicator)-2 ) //Iterar hacia adelante en "secondIndicator"
        //Buscar peaks confirmados (dos valores mayores antes y después) en "secondIndicator"
        if secondIndicator(i)>=secondIndicator(i-1) && secondIndicator(i-1)>=secondIndicator(i-2) &&
          secondIndicator(i)>=secondIndicator(i+1) && secondIndicator(i+1)>=secondIndicator(i+2)
                actualMinimum = -Inf //Inicializar mínimo actual como menos infinito
                if isempty( peaksPosition ) //Si no se han agregado peaks
                        lastPeak = 3 //Buscar hacia atrás hasta período tres
                else
                        lastPeak = peaksPosition(end)+1 //Buscar hacia atras hasta última divergencia
                end
                while (k = i, k>=lastPeak) //Buscar en "mainIndicator" hasta "lastPeak"
                        //Buscar peaks confirmados en "mainIndicator"
                        if mainIndicator(k)>=mainIndicator(k-1) && mainIndicator(k-1)>=mainIndicator(k-2) &&
                          mainIndicator(k)>=mainIndicator(k+1) && mainIndicator(k+1)>=mainIndicator(k+2)
                                if mainIndicator(k)>=actualMaximum
                                        peaksPosition(end+1) = k //Agregar div. confirmada en período "k"
                                        actualMaximum=mainIndicator(k) // Actualizar peak máximo actual
                                else //Si ya se agregaron los peaks
                                        break // Finalizar ciclo "k"
                                end
                        end
                        k=k-1 //Disminuir contador (recorrer en reversa)
                end
        end
        i=i+1 //Aumentar contador (recorrer hacia adelante)
end
```

Ilustración C.4: Pseudocódigo para extraer divergencias (funciones auxiliares <parte 4/5>)

getTripleDivergence

```
Divergence = getTripleDivergence( Divergence, Par )
TYPE={'regular','hidden'} //Patrones de divergencia
OPERATION={'long','short'} //Tipos de operación de divergencia
//Ciclo de todas las combinaciones
Loop OPERATION
        Loop TYPE
                CLEAR div
                //Estructura sólo con divergencias confirmadas
                div=Divergence.OPERATION.normal.confirmed.TYPE
                //Extraer divergencia triples confirmadas
                Divergence.OPERATION.triple.confirmed.TYPE=tripleDivergenceAux( div, Par)
                //Combinar divergencias confirmadas y parciales
                div = COMBINE( div , Divergence.OPERATION.normal.partial.TYPE )
                //Extraer divergencias triples parciales (considera div. parciales y confirmadas)
                Divergence.OPERATION.triple.partial.TYPE=tripleDivergenceAux( div, Par)
        end
end
```

tripleDivergenceAux

```
tripleDiv = tripleDivergenceAux( DivAux , Par)
Loop k through DivAux //Recorrer todas las divergencias en la estructura
        //Buscar si divergencia actual es continuada por otra divergencia
        firstDiv=find(DivAux .indicator.last(k)=DivAux .indicator.first)
        if ~isempty(firstDiv)
                //Períodos de duración de divergencia actual
                firstRange=DivAux .indicator.last(k)-DivAux .indicator.first(k)
                //Períodos de duración de divergencia continuada
                otherRange=DivAux .indicator.last(firstDiv)-DivAux .indicator.first(firstDiv)
                Loop j through otherRange //Ciclo por todas las divergencias continuadas
                        //Verificar si ambas divergencias poseen mismo rango temporal
                        if firstRange>=otherRange(j)/Par.maximumPeriodDiference &&
                          firstRange<=otherRange(j)*Par.maximumPeriodDiference
                                //Agregar divergencia triple a estructura
                                tripleDiv .indicator.first(end+1)=DivAux .indicator.first(k)
                                tripleDiv .indicator.middle(end+1)=DivAux .indicator.last(k)
                                tripleDiv .indicator.last(end+1)=DivAux .indicator.last(firstDiv(j))
                                tripleDiv .price.first(end+1)=DivAux .price.first(k)
                                tripleDiv .price.middle(end+1)=DivAux .price.last(k)
                                tripleDiv .price.last(end+1)=DivAux .price.last(firstDiv(j))
                        end
                end
        end
end
```

Ilustración C.5: Pseudocódigo para extraer divergencias (divergencias triples <parte 5/5>)

Anexo D: Pseudocódigo para extender variable dependiente

Entrada/Salida

Entrada
Close: Precio de cierre de instrumento financiero.
LongZZ: Vector con períodos de mínimos locales calculado con ZigZag.
ShortZZ: Vector con períodos de de máximos locales calculado con ZigZag.
Par: Estructura con parámetros.

Salida
Y: Estructura con períodos recomendados para compra y venta.

Par

Par.maximumPeriodPercentageExtension: Máxima extención porcentual de períodos entre ZigZags.
Par.maximumPricePercentageExtension: Máxima extención porcentual de precio entre ZigZags.
Par.earlyStop: Variable binaria que representa si se detiene la extensión tras un período excluído.
Par.maximumPeriodMean: Variable binaria que representa si se reemplaza extensión porcentual de períodos por mediana si se supera promedio.
Par.maximumPriceMean: Variable binaria que representa si se reemplaza extensión porcentualde precio por mediana si se supera promedio.

Inicialización

```
// Calcular promedio y mediana de cambio porcentual y períodos de diferencia por acción
Par.periodDifferenceMean = MEAN ( Combination of: LongZZ and Short ZZ )
Par.percentageChangeMean = MEAN ( PROFIT ( Combination of: LongZZ and Short ZZ ))
Par.percentageChangeMedian = MEDIAN ( PROFIT ( Combination of: LongZZ and Short ZZ ))
// Alcista
Loop k though LongZZ //Ciclo por todos los períodos alcistas
        Y.long=extendDependent(LongZZ(k), ShortZZ, Close, Par, “long”)
end
// Bajista
Loop k though ShortZZ //Ciclo por todos los períodos bajistas
        Y.short=extendDependent(Short(k), LongZZ, Close, Par, ”short”)
end
```

Ilustración D.1: Pseudocódigo para extender variable independiente (inicialización <parte 1/3>)

extendDependent

```
outVector = extendDependent( actualZZ, otherZZ, Close, Par, Type )
initialPrice=Close(actualZZ) //Precio de cierre de período actual de ZigZag
//Extensión hacia el futuro
nextIndex=find( otherZZ>actualZZ, 1 ) //Obtener primer ZigZag del otro tipo posterior a éste
if ~isempty(nextIndex) //Si no es el último ZigZag
        forwardPeriodDifference=otherZZ(nextIndex)-actualZZ //Períodos hasta siguiente señal
        if Par.maximumPeriodMean //Verificar si se acota la diferencia de períodos por el promedio
                forwardPeriodDifference=min(forwardPeriodDifference,Par.periodDifferenceMean)
        end
        forwardPercentageChange=(Close(nextIndex)-initialPrice)/initialPrice //Cambio porcentual futuro
        if forwardPercentageChange>Par.percentageChangeMean //Cambio porcentual sobre proemdio
                forwardPercentageChange=Par.percentageChangeMedian //Cambiar por mediana
        end
else // Si es el último ZigZag
        forwardPeriodDifference=Par.periodDifferenceMean //Períodos de diferencia como promedio
        forwardPercentageChange=Par.percentageChangeMedian //Cambio porcentual como mediana
end
//Calcular límite de períodos y cambio porcentual
maximumPeriods=ceil(forwardPeriodDifference*Par.maximumPeriodPercentageExtension)
maximumPercentage=forwardPercentageChange*Par.maximumPricePercentageExtension
// Ciclo hacia adelante hasta los períodos calculados
for t = actualZZ:min(actualZZ+maximumPeriods, length(Close) )
        // Verificar condición, agregar período "t" como positivo y verificar glad de finalización de ciclo
        [outVector,flag] = checkCondition(outVector, Close(t), Type, initialPrice, maximumPercentage, Par)
        if flag
                break // Finalizar ciclo
        end
end
//Extensión hacia el pasado
beforeIndex=find( otherZZ<actualZZ, 1, "last") //Obtener primer ZigZag del otro tipo anterior a éste
if ~isempty(beforeIndex) //Si no es el primer ZigZag
        backwardPeriodDifference=actualZZ-otherZZ(beforeIndex) //Períodos desde la última señal
        backwardPercentageChange=(Close(beforeIndex)-initialPrice)/initialPrice //Cambio porcentual
else //Si este período es el primer ZigZag
        backwardPeriodDifference=periodDifferenceMean //Períodos de diferencia como promedio
        backwardPercentageChange=percentageChangeMedian //Cambio porcentual como mediana
end
//Calcular límite de períodos y cambio porcentual
maximumPeriods=ceil(backwardPeriodDifference*Par.maximumPeriodPercentageExtension)
maximumPercentage=backwardPercentageChange*Par.maximumPricePercentageExtension
// Ciclo hacia atras hasta los períodos calculados
initialPeriod=max(1, actualZZ-maximumPeriods)
for t = initialPeriod:actualZZ
        // Verificar condición, agregar período "t" como positivo y verificar glad de finalización de ciclo
        [outVector,flag] = checkCondition(outVector, Close(t), Type, initialPrice, maximumPercentage, Par)
        if flag
                break // Finalizar ciclo
        end
end
```

Ilustración D.2: Pseudocódigo para extender variable independiente (principal <parte 2/3>)

checkContidion

```
[outVector, flag] = checkCondition(outVector, actualClose, Type, initialPrice, maximumPercentage, Par)
flag = false // inicialización como falsa
switch Type // Verificar si se está en alza o baja
        case “long” // Estados alcista
                //Verificar precio bajo máximo cambio porcentual
                if actualClose<=initialPrice*(1+maximumPercentage)
                        outVector(t) = 1 //Etiquetar período como positivo
                elseif Par.earlyStop // Si no, se activa “earlyStop”
                        flag=true
                end
        case “short” // Estado bajista
                //Verificar precio sobre máximo cambio porcentual
                if actualClose>=initialPrice*(1-maximumPercentage)
                        outVector(t) = 1 //Etiquetar período como positivo
                elseif Par.earlyStop // Si no, se activa “earlyStop”
                        flag=true
                end
end
```

Ilustración D.3: Pseudocódigo para extender variable independiente (función auxiliar <parte 3/3>)

Anexo E: Pseudocódigo simulador de inversiones

Estrada/Salida

Entrada
Close, High, Low: Precios OHL de instrumento financiero.
Dataset: Predicciones de compraventa para cada empresa.
Companies: Empresas a aplicar simulación.
Par: Estructura de parámetros.

Salida
tradingInfo: Estructura con información de simulación.

Par

Par.windowSize: Número de períodos a considerar en el pasado.
Par.ammountOpenPosition: Número mínimo de señales para abrir operación.
Par.ammountClosePosition: Número mínimo de señales para cerrar operación.
Par.maximumUtilizationOpen: Número máximo de veces que una señal puede ser utilizada para abrir op.
Par.maximumUtilizationClose: Número máximo de veces que una señal puede ser utilizada para cerrar op.
Par.numberOperationsClose: Número de oepraciones cerradas a la vez mediante heurística.
Par.maximumPosition: Número máximo de operacioens abiertas simultáneamente (para compra y venta).
Par.hedging: Variable binaria, si se pueden abrir oepracioens de compra y venta a la vez.
Par.instantClose: Variable Binaria, si SL y TP se aplica a cruce de *High*/*Low* o a precio de cierre.
Par.stopLossLong: SL porcentual estatico aplicado a operaciones de compra.
Par.takeProfitLong: TP porcentual estatico aplicado a operaciones de compra.
Par.stopLossShort: SL porcentual estatico aplicado a operaciones de venta.
Par.takeProfitShort: TP porcentual estatico aplicado a operaciones de venta.

Inicialización

```
tradingInfo = EMPTY STRUCT // Se define "tradingInfo" como estructura vacía.
If inmediateClose // Si SL y TP se aplica antes que período finalice, se acota DD y retorno
        maxDD.long = Par.stopLossLong
        maxDD.short = Par.stopLossShort
        maxR.long = Par.takeProfitLong
        maxR.short = Par.takeProfitShort
else // Si no se a plica a período de cierre, no posee cotas
        maxDD.long = Inf
        maxDD.short = Inf
        maxR.long = Inf
        maxR.short = Inf
end
Loop k through all conpanies //Simular y extraer "tradingInfo" para todas las empresas
        tradingInfo = simulate( Dataset.company(t), tradingInfo, Par, maxDD, maxR)
end
```

Ilustración E.1: Pseudocódigo simulador (inicialización <parte 1/5>)

simulate

```
tradingInfo = simulate( data ,  tradingInfo, Par, maxDD, maxR  )
periods = number of periods in data
//Inicializar variables de control
openLongPeriod = EMPTY SRUCT
openShortPeriod = EMPTY SRUCT
//Inicializar contadores de máxima utilización de señales para abrir y cerrar operaciones
usedLongOpenPeriod = zeros( period,1 ) // Inicializar con Ceros (No se han utilizado señales)
usedLongClosePeriod = zeros( period,1 ) // Inicializar con Ceros (No se han utilizado señales)
usedShortOpenPeriod = zeros( period,1 ) // Inicializar con Ceros (No se han utilizado señales)
usedShortClosePeriod = zeros( period,1 ) // Inicializar con Ceros (No se han utilizado señales)
// Transferir señales de compraventa (Utilizado en uso máximo de predicciones)
openLongPredictor = data.predictor.long // Inicializar como predictor alcista original
closeLongPredictor = data.predictor.long // Inicializar como predictor alcista original
openShortPredictor = data.predictor.short // Inicializar como predictor bajista original
closeShortPredictor = data.predictor.short // Inicializar como predictor bajista original
Loop t =1 to periods // Recorrer todos los períodos de la acción
        //Obtener número de operaciones de compra y venta para abrir y cerrar operaciones (heurística)
        windowStart = max(t-Var.windowSize+1, 1) // Período de comienzo de ventana
        timeWindow = windowStart:t // Períodos de la ventana
        longWindowOpen = sum( openLongPredictor(timeWindow) ) // # de señales en ventana (compra)
        longWindowClose = sum( closeLongPredictor(timeWindow) ) //  # de señales en ventana (compra)
        shortWindowOpen = sum( openShortPredictor(timeWindow) ) //  # de señales en ventana (venta)
        shortWindowClose = sum( closeShortPredictor(timeWindow) ) //  # de señales en ventana (venta)
        // Verificar cierre de operaciones mediante SL y TP
        [openLongPeriod , orderInfo] = loopClosePositionBySLTP(openLongPeriod , orderInfo, t , data, ...
                                                                Par, 'L', maxDD.long, maxR.long)
        [openShortPeriod , orderInfo] = loopClosePositionBySLTP(openShortPeriod , orderInfo, t , data, ...
                                                                Par, 'S', maxDD.short, maxR.short)
        // Verificar cierre de operacioens mediante heurística
        [openLongPeriod, usedLongClosePeriod, closeLongPredictor, orderInfo]=
        closePositionsByHeuristic( openLongPeriod , orderInfo, t, data,  Par, 'L', maxDD.long, maxR.long,...
                                    Par.maxRlong, usedLongClosePeriod, closeLongPredictor, timeWindow )
        [openShortPeriod, usedShortClosePeriod, closeShortPredictor, orderInfo]=
        closePositionsByHeuristic( openShortPeriod , orderInfo, t, data,  Par, 'S', maxDD.short, maxR.short,...
                                    Par.maxRshort, usedShortClosePeriod, closeShortPredictor, timeWindow )
        // Verificar apertura de operaciones mediante heurística
        [openLongPeriod, usedLongOpenPeriod, openLongPredictor,  orderInfo] =
        closePositionsByHeuristic( openLongPeriod, orderInfo, t, data,  Par, 'L', ...
                                    usedLongOpenPeriod, openLongPredictor, timeWindow)
        [openShortPeriod, usedShortOpenPeriod, openShortPredictor,  orderInfo] =
        closePositionsByHeuristic( openShortPeriod, orderInfo, t, data,  Par, 'L', ...
                                    usedShortOpenPeriod, openShortPredictor, timeWindow)
end
```

Ilustración E.2: Pseudocódigo simulador (ciclo principal <parte 2/5>)

loopClosePositionBySLTP

```
[operationVector, orderInfo] = loopClosePositionBySLTP(operationVector, orderInfo, actualPeriod, data, ...
                                                       Par, operationType, maxDD, maxR)
if ~isempty( operationVector) // Si hay operaciones abiertas
        // Verificar cierre mediante SL y TP
        nOp = length( operationVector) // Número de operaciones abiertas
        Loop op = 1 to nOp //Recorrer todas las operaciones abiertas
                opi = nOpLong – op + 1 // Ciclo inverso a través de operaciones
                tradingInfo = checkCloseBySLTP( openLongPeriod, opi, actualPeriod, data,
                                                tradingInfo, Par, operationType, maxDD, maxR)
        end
end
```

closePosition

```
[OperationVector, tradingInfo] = closePosition(operationVector, positionCloseIndex, closePrice,...
                                 tradingInfo, operationType, data, actualTime, closeReaon, maxDD, maxR)
tradingInfo.openTime(end+1)=positionVector(positionCloseIndex) // Agregar período de apertura
tradingInfo.closeTime(end+1)=actualTime // Agregar período de cierre
openPrice=Close(tradingInfo.openTime(end)) // Precio de apertura
tradingInfo.long(end+1) =  ( operationType =='L' ) // Agregar si tipo de orden es alcista
tradingInfo.short(end+1) = ( operationType =='S' ) // Agregar si tipo de orden bajista
switch operationType //Depende del tipo de operación
        case 'L'  // Operación de compra
                // Obtener precio mínimo mientras la oepracione se encontraba abierta
                minPrice=min(Low(tradingInfo.openTime(end):tradingInfo.closeTime(end)))
                retorno=(closePrice-openPrice)/openPrice //Retorno de compra
                // Drawdown no puede superar maxDD (depende de Par.inmediateClose)
                tradingInfo.porcentualDD(end+1) =min(maxDD, (openPrice-minPrice)/openPrice )
        case 'S'  // Operación de venta
                // Obtener precio máximo mientras la oepracione se encontraba abierta
                maxPrice=max(High(tradingInfo.openTime(end):tradingInfo.closeTime(end)))
                retorno=(openPrice-closePrice)/closePrice //Retorno de venta
                // Drawdown no puede superar maxDD (depende de Par.inmediateClose)
                tradingInfo.porcentualDD(end+1) =min(maxDD, (maxPrice-openPrice)/openPrice)
End
If retorno>0 //Retorno positivos acotado por maxR (depende de Par.inmediateClose)
        tradingInfo.return(end+1) = min( maxR, retorno ) // No puede superar maxR
else //Retorno negativo acotado por -maxDD (depende de Par.inmediateClose)
        tradingInfo.return(end+1) = max( -maxDD, retorno ) // No puede ser menor a -maxDD
end
tradingInfo.duration(end+1)=tradingInfo.closeTime(end)-tradingInfo.openTime(end) // Duración de op.
tradingInfo.closeReason.sl(end+1) = ( closeReason=='SL' ) //Información sobre razón de cierre
tradingInfo.closeReason.tp(end+1) =  ( closeReason=='TP' ) //Información sobre razón de cierre
tradingInfo.closeReason.heuristic(end+1) =  ( closeReason=='Heuristic' ) //Información sobre razón de cierre
OperationVector = DELETE POSITION( positionCloseIndex ) //Borrar de vector con operaciones abiertas
```

Ilustración E.3: Pseudocódigo simulador (ciclo cierre posiciones mediante SL y guardado <parte 3/5>)

checkCLoseBySLTP

```
tradingInfo = checkCloseBySLTP( operationVector, orderIndex, actualPeriod, data, tradingInfo, Par, ...
                                operationType, maxDD, maxR)
closePrice = -1 // Inicializar negativo (no cerrar operaciones)
switch operationType // Depende del tipo de operación
        case 'L' // Verificar SL y TP para operaciones de compra
        SL = Close(operationVector(orderIndex))*(1-Par.stopLossLong) // Precio de SL para compra
        TP = Close(operationVector(orderIndex))*(1+Par.takeProfitLong) // Precio de TP para compra
        If Par.inmediateClose //Primero verificar si máximo o mínimo activan SL, luego TP
                if Low(actualperiod) <= SL // Cerrar por activacion de SL
                        closePrice = SL // Precio actual de cierre
                        closeReason= 'SL' // Razón de cierre
                elseif High(actualperiod) >= TP // Cerrar por activacion de TP
                        closePrice = TP // Precio actual de cierre
                        closeReason= 'TP' //Razón de cierre
                end
        else //Verificar si precio de cierre activa SL o TP
                if Close(actualperiod) <= SL // Cerrar por activacion de SL
                        closePrice = Close(actualperiod) // Precio actual de cierre
                        closeReason= 'SL' //Razón de cierre
                elseif Close(actualperiod) >= TP // Cerrar por activacion de TP
                        closePrice = Close(actualperiod) // Precio actual de cierre
                        closeReason= 'TP' // Razón de cierre
                end
        end
        case 'S' // Verificar SL y TP para operaciones de venta
        SL = Close(operationVector(orderIndex))*(1+Var.stopLossShort) // Precio de SL para venta
        TP = Close(operationVector(orderIndex))*(1-Var.takeProfitShort) // Precio de TP para venta
        If Par.inmediateClose //Primero verificar si máximo o mínimo activan SL, luego TP
                if High(actualperiod) >= SL // Cerrar por activacion de SL
                        closePrice = SL // Precio actual de cierre
                        closeReason= 'SL' // Razón de cierre
                elseif Low(actualperiod) >= TP // Cerrar por activacion de SP
                        closePrice = TP // Actual price close
                        closeReason= 'TP' // Razón de cierre
                end
        else // Check Close price reaches SL and TP
                if Close(actualperiod) >= SL // Cerrar por activacion de SL
                        closePrice = Close(actualperiod) // Precio actual de cierre
                        closeReason= 'SL' // Razón de cierre
                elseif Close(actualperiod) <= TP // Cerrar por activacion de TP
                        closePrice = Close(actualperiod) // Precio actual de cierre
                        closeReason= 'TP' // Razón de cierre
                end
        end
end
If closePrice>0 // Si se activa SL o TP
        [operationVector,tradingInfo ] = closePosition(operationVector, orderIndex, closePrice , ...
                                tradingInfo, operationType, data, actualPeriod, closeReason, maxDD)
end
```

Ilustración E.4: Pseudocódigo simulador (chequeo SL y TP <parte 4/5>)

openPositionsByHeuristic

```
[operationVector, usedOpenPeriod, openPredictor,  orderInfo] = closePositionsByHeuristic( ...
                                operationVector, orderInfo, actualPeriod, data,  Var, operationType, ...
                                                        maxDD, usedOpenPeriod, openPredictor, timeWindow)
 //Si Aún no se alcanza máximo número de operaciones abiertas
if  length(operationVector)<.maximumPosition
        // No se puede abrir otra operación si: (~Par.hedging && existOtherTypeOfOperation)
        if noOtherOperation || Var.hedging
                operationVector(end+1)=actualPeriod
        end
        if Var.maximumUtilizationOpen //Si hay límite de recomendaciones en apertura de op.
                //Obtener índice de recomendaciones utilizadas
                usedIndex=find(openPredictor(timeWindow)>0)+windowStart-1
                usedIndex=usedIndex(1:min(length(usedIndex,Var.ammountOpenPosition)))
                //Aumentar contador de recomendaciones utilizadas
                usedOpenPeriod(usedIndex )= usedOpenPeriod(usedIndex) + 1
                //Desabilitar recomendaciones que sobrepasen parámetro de umbral
                openPredictor( usedOpenPeriod>=Var.ammountClosePosition ) = 0
        end
end
```

closePositionsByHeuristic

```
[operationVector, usedClosePeriod, closePredictor,  orderInfo] = closePositionsByHeuristic(...
                                operationVector, orderInfo, actualPeriod, data,  Par, operationType, ...
                                maxDD, maxR, usedClosePeriod, closePredictor, timeWindow)
if ~isempty( operationVector) // Si hay operaciones abiertas
        if windowClose>=Par.ammountClosePosition // Verificar si hay suficientes señales para cierre
                // Cerrar operaciones hasta un límite de Par.numberOperationsClose
                closeNumber = min( length(operationVector), Var.numberOperationsClose  )
                Loop op = 1 to closeNumber // Ciclo de número de operaciones a cerrar
                        // Cerrar a partir de la última (La primera abierta)
                        [operationVector,tradingInfo ] = closePosition(operationVector, 1, ...
                                Close(actualPeriod) ,  tradingInfo, operationType, data, ...
                                actualPeriod, 'Heuristic', maxDD, maxR)
                end
                if Par.maximumUtilizationClose //Si hay límite de recomendaciones en cierre de op.
                        //Obtener índice de recomendaciones utilizadas
                        usedIndex=find(closePredictor(timeWindow)>0)+windowStart-1
                        usedIndex=usedIndex(1:min(length(usedIndex,Var.ammountClosePosition)))
                        //Aumentar contador de recomendaciones utilizadas
                        usedClosePeriod(usedIndex) = usedClosePeriod(usedIndex) + 1
                        //Desabilitar recomendaciones que sobrepasen parámetro de umbral
                        closePredictor( usedClosePeriod>=Par.ammountClosePosition ) = 0
                end
        end
end
```

Ilustración E.5: Pseudocódigo simulador (abrir y cerrar por heurística <parte 5/5>)

Anexo F: Detalles de análisis de ZigZag

RP de ZigZag	Períodos de diferencia [Días]		Cambio porcentual [%]	
	Promedio	Mediana	Promedio	Mediana
2%	7.2284	5	7.31	5.16
3%	10.65	7	9.74	6.92
4%	14.6469	10	12.24	8.63
5%	19.6688	13	15.18	11.01
6%	26.0204	17	18.68	13.81
7%	30.9987	21	21.32	16.06
8%	36.0332	24	23.77	17.8
10%	44.9795	33	28.38	22.39
12%	52.4593	36	32.53	26.03
14%	60.3066	45	36.81	28.35

Tabla F.1: Períodos de diferencia vs. RP de ZigZag (base general)

RP de ZigZag	Períodos de diferencia [Días]		Cambio porcentual [%]	
	Promedio	Mediana	Promedio	Mediana
2%	7.2818	5	7.44	5.26
3%	10.8284	6	9.93	7.34
4%	14.8293	9	12.55	9.07
5%	20.2633	13	15.59	11.46
6%	27.7283	16	19.27	14.5
7%	33.4737	21	21.95	16.5
8%	38.9164	22	24.36	18.06
10%	44.7438	27	27.37	21.82
12%	52.2	32	31.16	23.66
14%	60.0492	43	34.45	0.2701

Tabla F.2: Períodos de diferencia vs. RP de ZigZag (base alcista)

RP de ZigZag	Períodos de diferencia [Días]		Cambio porcentual [%]	
	Promedio	Mediana	Promedio	Mediana
2%	7.176	5	7.18	5.07
3%	10.4761	8	9.56	6.65
4%	14.4711	10	11.95	8.17
5%	19.1043	13	14.8	10.55
6%	24.4295	17	18.14	12.82
7%	18.7367	21	20.74	15.25
8%	33.4444	26	23.23	17.65
10%	45.1822	35	29.24	22.94
12%	52.672	37	33.66	27.89
14%	60.5132	47	38.71	29.76

Tabla F.3: Períodos de diferencia vs. RP de ZigZag (base bajista)

Anexo G: Matrices de confusión de entrenamiento balanceado

Clase predicha	Clase real	
	Si	No
Si	1153	558
No	149	744

Tabla G.1: Matriz de confusión de BPNN en base alcista

Clase predicha	Clase real	
	Si	No
Si	1144	575
No	158	727

Tabla G.2: Matriz de confusión de SVM en base alcista

Clase predicha	Clase real	
	Si	No
Si	869	315
No	433	987

Tabla G.3: Matriz de confusión de SBM en base alcista

Clase predicha	Clase real	
	Si	No
Si	1078	597
No	266	747

Tabla G.4: Matriz de confusión de BPNN en base bajista

Clase predicha	Clase real	
	Si	No
Si	1026	546
No	318	798

Tabla G.5: Matriz de confusión de SVM en base bajista

Clase predicha	Clase peal	
	Si	No
Si	1046	561
No	298	783

Tabla G.6: Matriz de confusión de SBM en base bajista

Anexo H: Detalle de resultado entrenamiento sobre base no balanceada

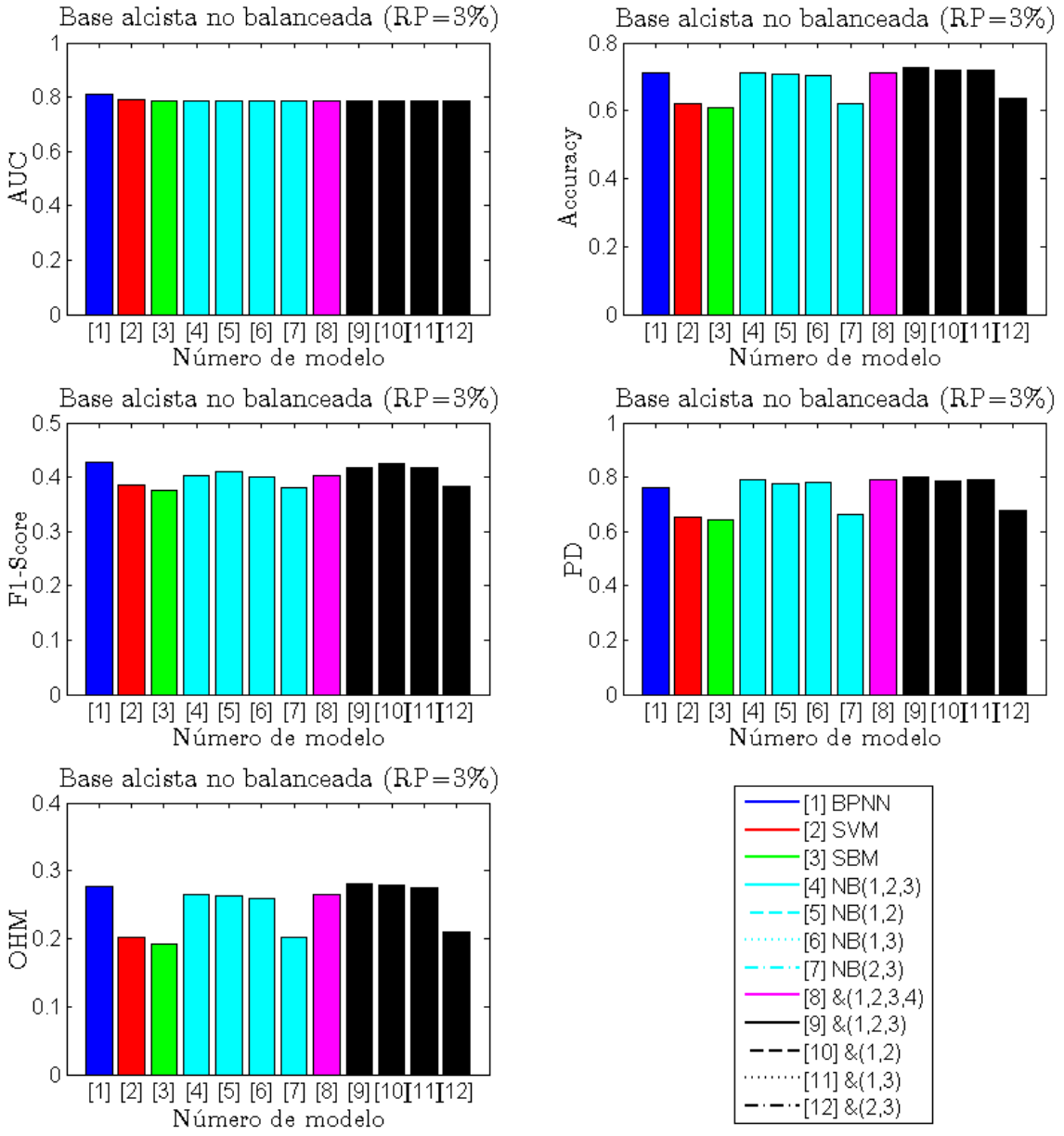

Ilustración H.1: Resultado de entrenamiento sobre base no balanceada alcista

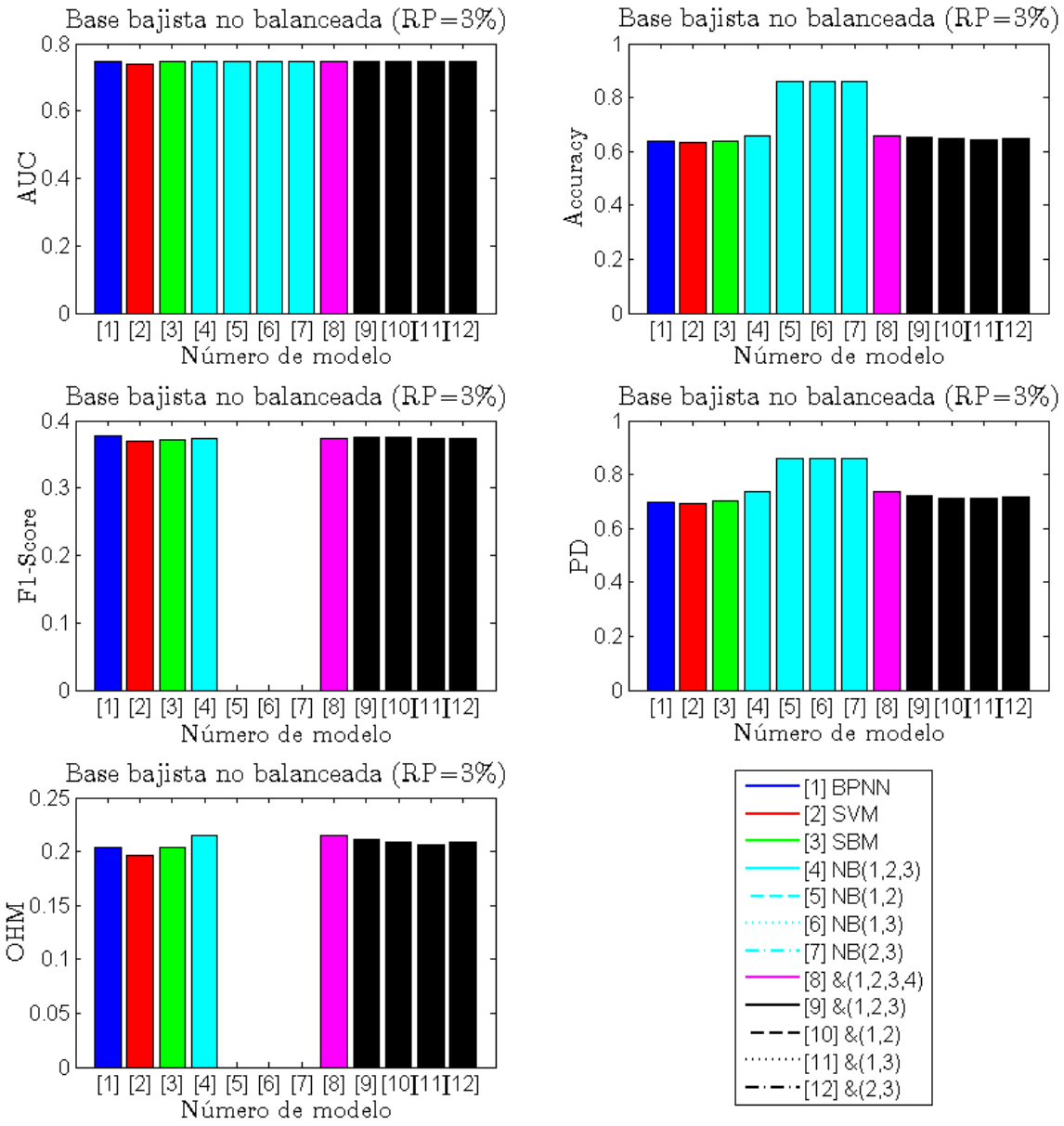

Ilustración H.2: Resultado de entrenamiento sobre base no balanceada bajista

Anexo I: Recomendaciones de compraventa sobre base de testeo

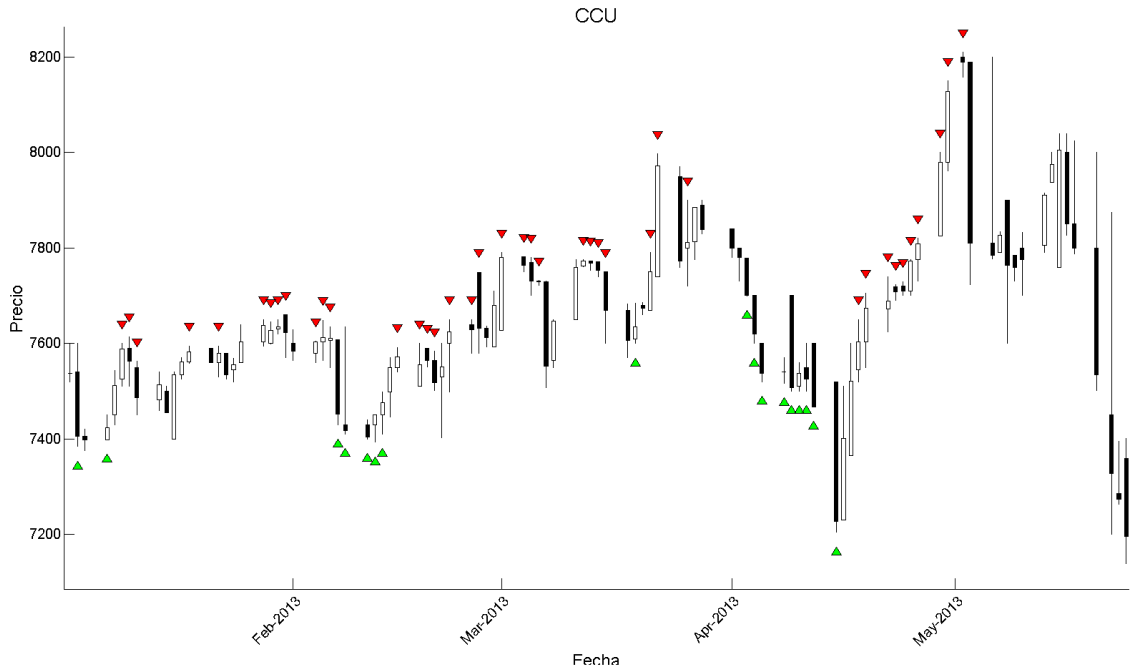

Ilustración I.1: Resultados predicción aplicado a CCU (períodos de testeo)

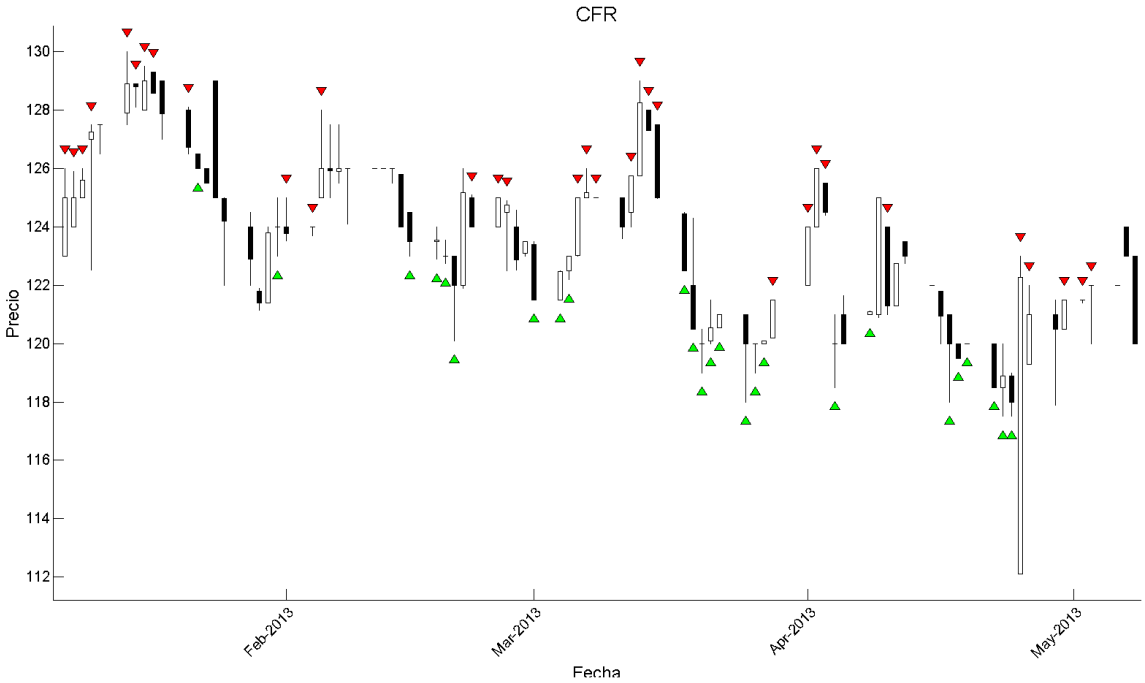

Ilustración I.2: Resultados predicción aplicado a CFR (períodos de testeo)

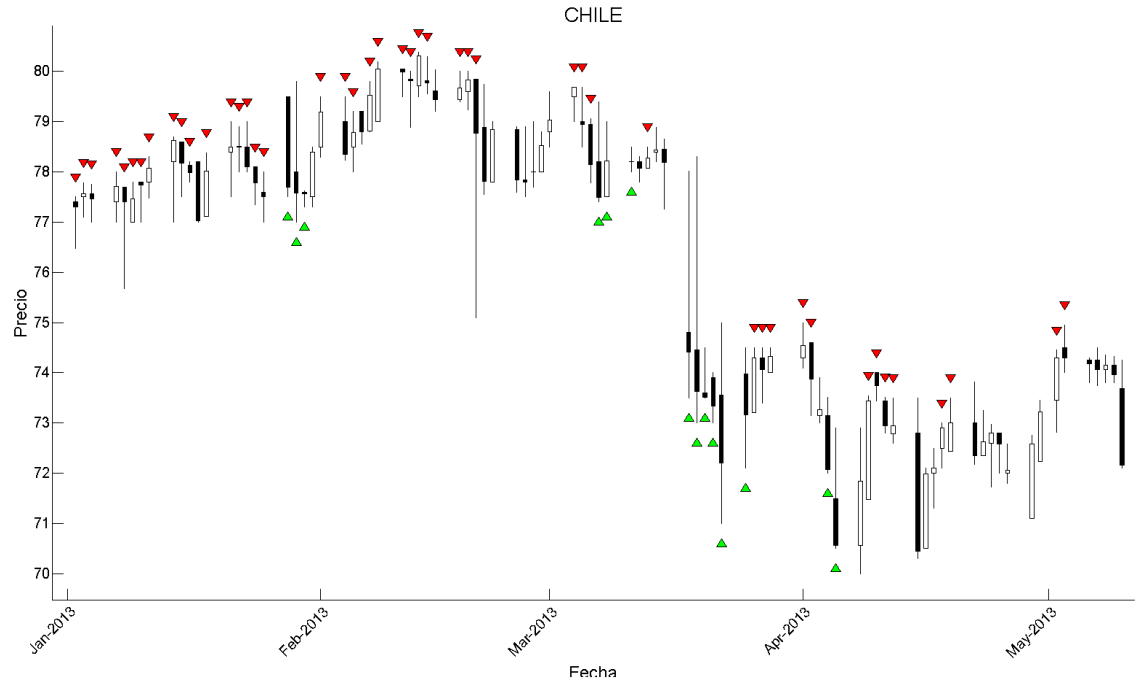

Ilustración I.3: Resultados predicción aplicado a CHILE (períodos de testeo)

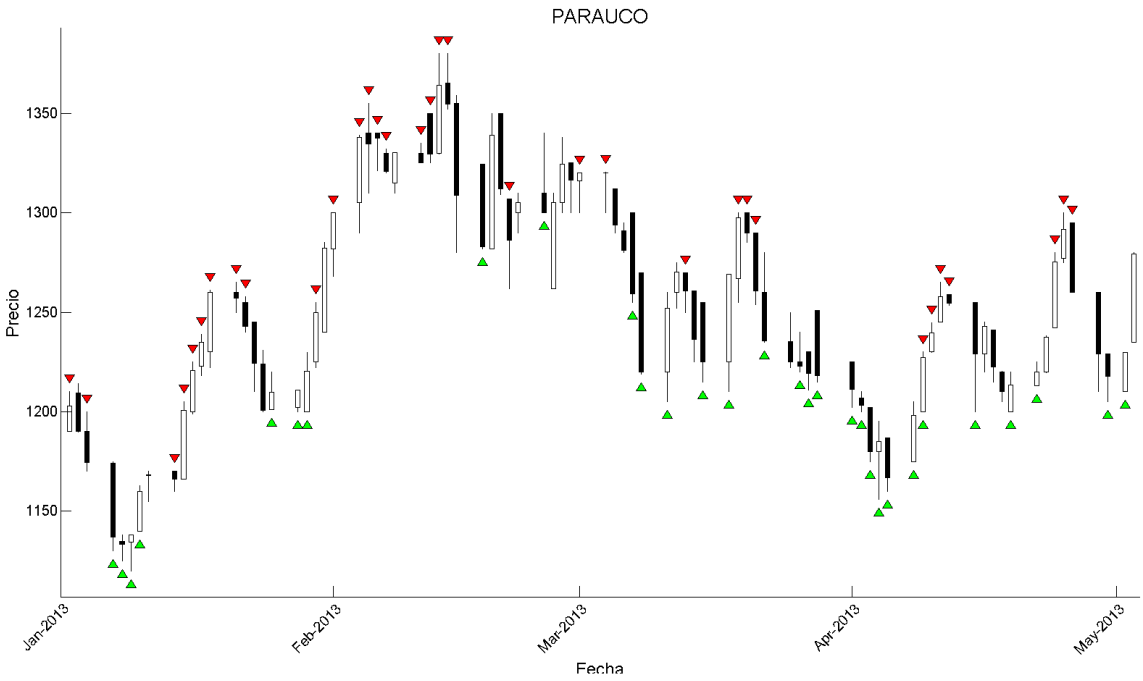

Ilustración I.4: Resultados predicción aplicado a PARAUCO (períodos de testeo)

Anexo J: Resultados configuraciones de simulación

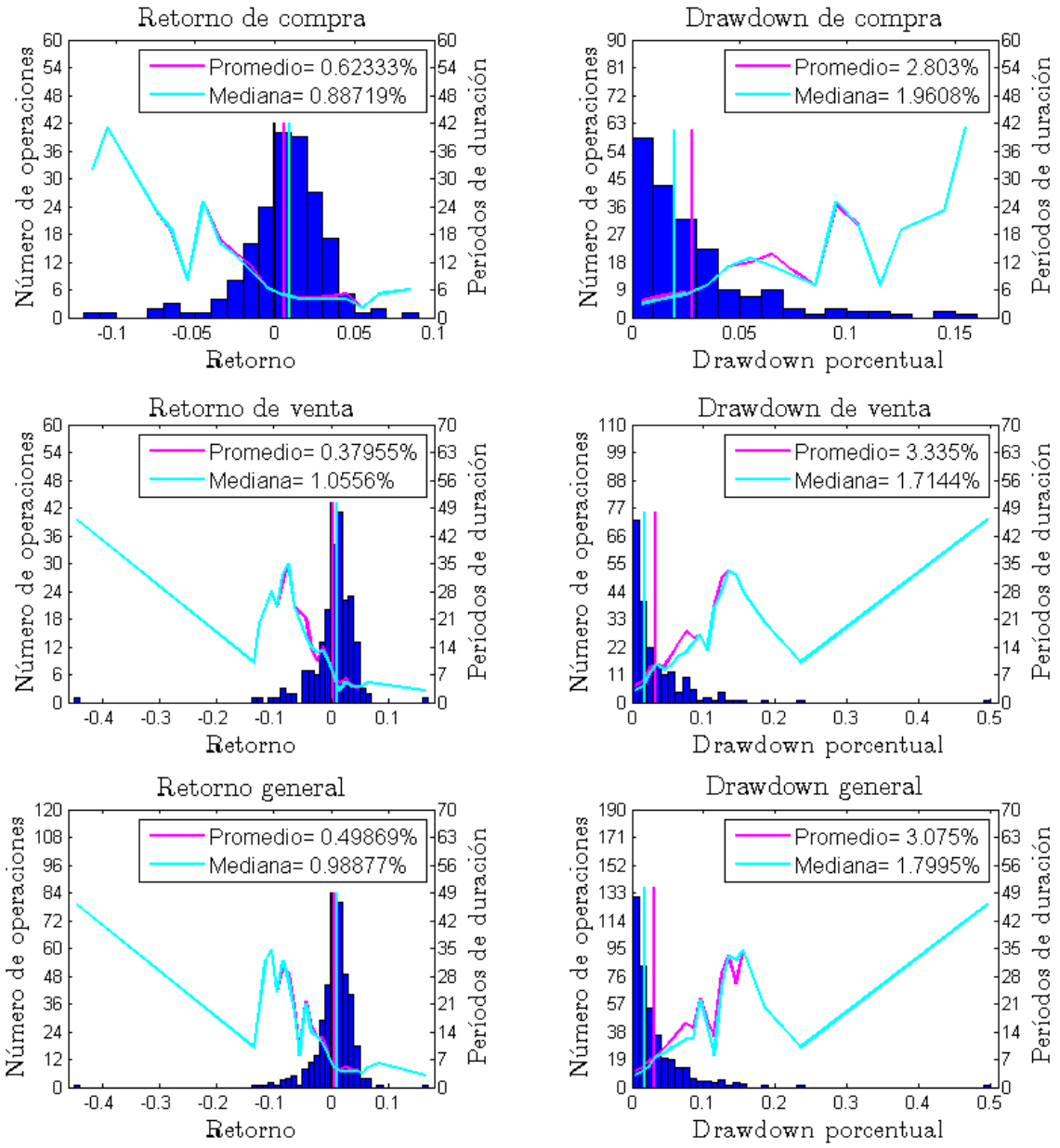

Ilustración J.1: Retorno y *drawdown* para compra, venta y general de simulador de transacciones aplicado a configuración uno

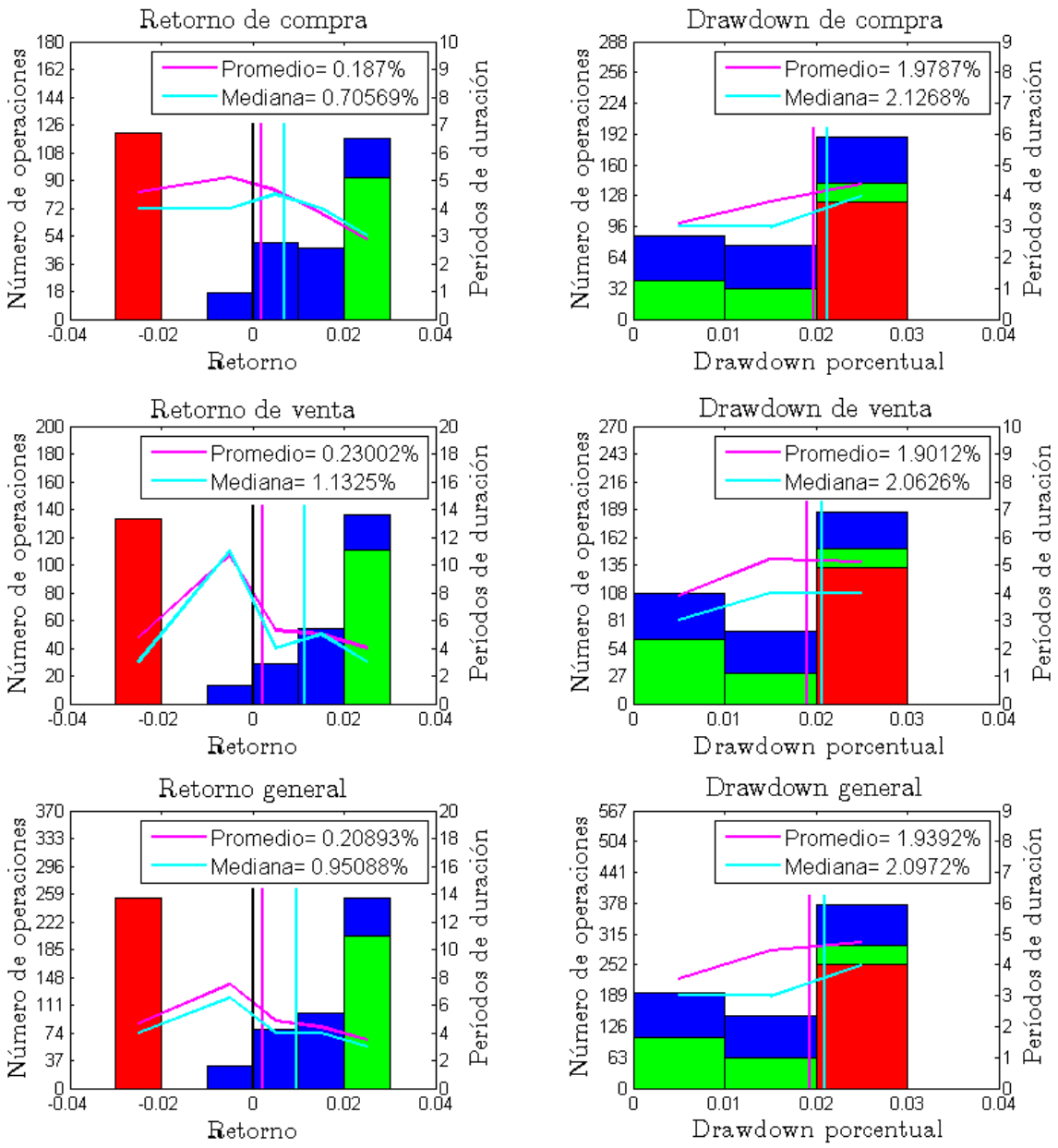

Ilustración J.2: Retorno y *drawdown* para compra, venta y general de simulador de transacciones aplicado a configuración dos

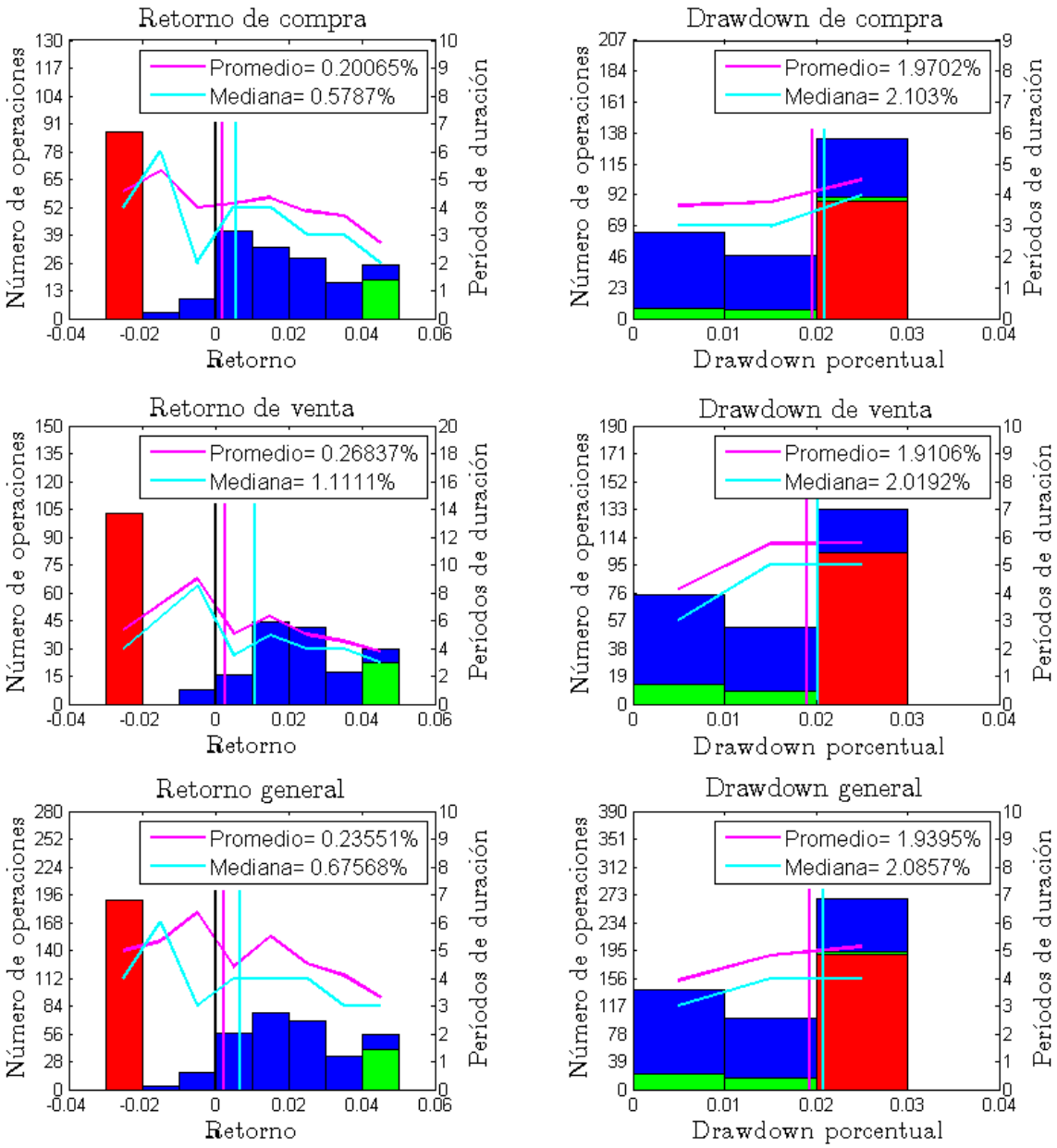

Ilustración J.3: Retorno y *drawdown* para compra, venta y general de simulador de transacciones aplicado a configuración tres

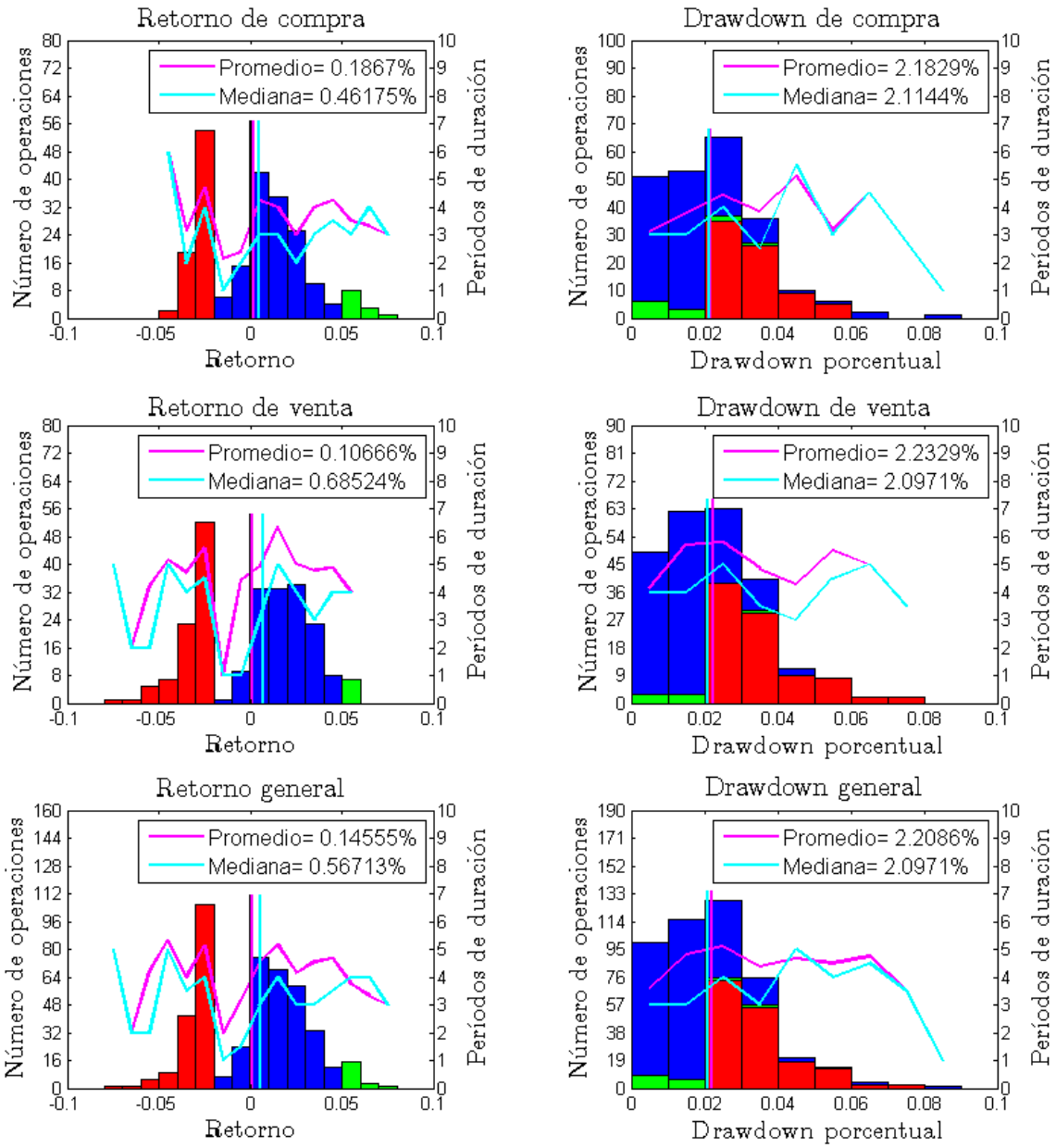

Ilustración J.4: Retorno y *drawdown* para compra, venta y general de simulador de transacciones aplicado a configuración cuatro

Anexo K: Esquema de proceso general

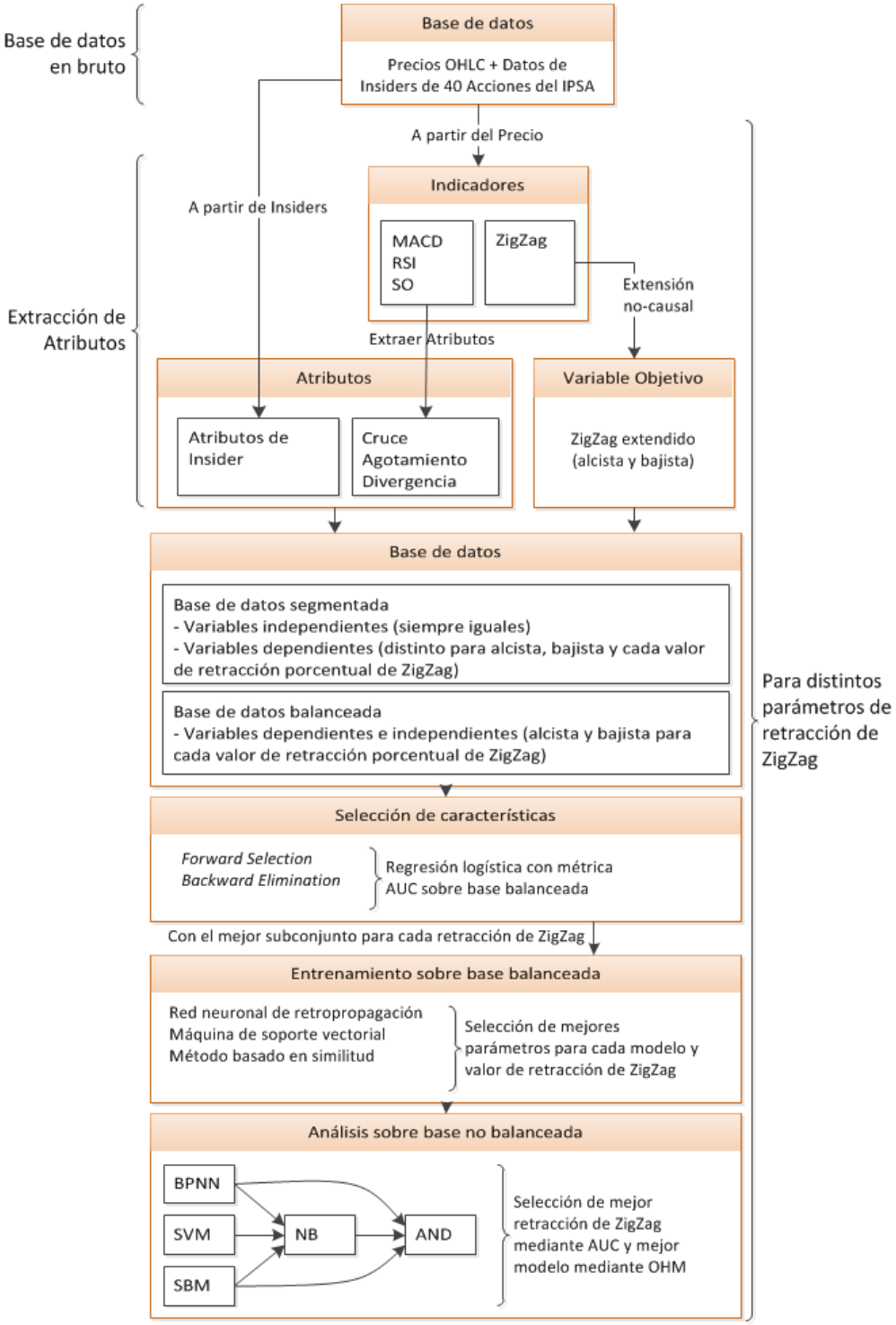

Ilustración K.1: Diagrama esquemático de proceso general