

INFORME SOBRE EL INDICADOR i22

por Rafael Pinter (rafaelpinter@gmail.com)

Introducción

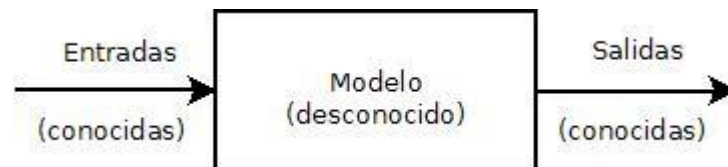
El presente informe pretende describir brevemente el funcionamiento del indicador i22. El objetivo de dicho indicador es obtener modelos a partir de datos conocidos. Posteriormente, se hace una evaluación del mismo, aplicándolo a un problema concreto: operaciones bursátiles a corto plazo.

A continuación, se describe brevemente el funcionamiento del indicador. Después se presentan los resultados obtenidos en la aplicación al caso concreto mencionado. Finalmente se compara con los resultados obtenidos por algunos indicadores del Análisis Técnico (AT), operando en condiciones similares.

1. Breve descripción del indicador i22

Lo que se diga sobre el indicador i22 es también válido para i21, ya que ambos son totalmente compatibles en sus entradas y salidas, teniendo únicamente pequeñas diferencias en el código.

El objetivo de i22 es encontrar un modelo a partir de unos datos de entrada y salida conocidos. El problema se representa en la siguiente figura:



Se puede intentar obtener un modelo para realizar operaciones bursátiles a largo, medio o corto plazo, o se podría intentar optimizar una cartera. Para realizar el aprendizaje se pueden usar series temporales de precios (precios diarios, semanales, etc), o utilizar datos procedentes del Análisis Fundamental, o de la coyuntura económica, según el objetivo buscado.

Para obtener el modelo, i22 utiliza metodologías de la Inteligencia Artificial (IA). El proceso a seguir será el siguiente:

1. En primer lugar, se realiza un proceso de entrenamiento, sobre un conjunto de datos (conjunto de aprendizaje) que contiene datos de entrada y sus correspondientes salidas “correctas” (conocidas). El resultado de este proceso de entrenamiento es una población de individuos, cada uno de los cuales es un posible modelo del problema que queremos analizar.
2. En segundo lugar, hay que validar los individuos obtenidos, para seleccionar, si ello es posible, aquéllos que pueden ser buenos modelos del problema. La validación se

realiza haciendo que, esta vez, sea el modelo el que, para cada dato de entrada, proponga un dato de salida. Las salidas propuestas se comparan con las salidas “correctas”, que también conocemos y, de esta forma, se puede hacer una estimación de la calidad de cada individuo. El conjunto que utilizamos (conjunto de validación) es un conjunto disjunto con respecto al conjunto de aprendizaje.

3. Finalmente, los individuos que hayan demostrado su calidad en el proceso de validación (si existen), podrán ser considerados, en principio, buenos modelos del problema que queríamos modelar.

En lo que sigue vamos a aplicar lo dicho anteriormente a un problema concreto: el intento de obtener un modelo para el mercado bursátil de un ticker, con el objetivo de realizar operaciones de unos pocos días de duración, utilizando como datos para realizar el aprendizaje la serie temporal diaria de ese ticker. Para obtener algún resultado medible, cambiaremos el punto 3 anterior de la siguiente forma: se evaluarán los individuos seleccionados como buenos modelos. Para ello se considera un nuevo conjunto de datos (conjunto de evaluación) y se hace que sea el modelo el que, para cada dato de entrada, proponga un dato de salida. Las salidas propuestas se comparan con las salidas “correctas”, que también conocemos y, de esta forma, se evalúa cada individuo, tal como se explica más adelante. El conjunto que utilizamos (conjunto de evaluación) es un conjunto disjunto con respecto al conjunto de aprendizaje y con respecto al conjunto de validación.

Consideramos un modelo (M_{i22}) como un sistema que recibe unos datos de entrada y genera unos datos de salida. Los datos de entrada (x_t) son los proporcionados por la serie temporal en t , y el dato de salida (p_t) es la propuesta del modelo, obtenida en t , para $t+1$, $p_t = M_{i22}(x_t)$. La propuesta es un valor entre -1.0 y +1.0, correspondiendo el valor -1.0 a la mayor intensidad de “posición corta” y el valor +1.0 a la mayor intensidad de “posición larga”. El objetivo es optimizar el modelo, conociendo los datos de entrada y los datos de salida, para que sea capaz de generalizar, y obtener propuestas para datos de entrada no explorados previamente.

Como datos de entrada se utilizan los valores diarios, para un determinado ticker, del precio de apertura de la sesión (Open), el precio del cierre de la sesión (Close), el precio máximo alcanzado en la sesión (High), el precio mínimo alcanzado en la sesión (Low) y el volumen negociado en la sesión (Vol). Estos datos son preprocesados para obtener las variables de entrada al sistema. Los datos de salida se obtienen a partir de estos mismos datos mediante un proceso que se describe más adelante. Estos datos, ordenados por fecha, se agrupan en varios conjuntos disjuntos.

Sin entrar en detalles, el proceso seguido se describe a continuación.

Tenemos una población de individuos. Estos individuos sufren un proceso de aprendizaje, cuyo objetivo es obtener un modelo del funcionamiento del mercado.

El conjunto de aprendizaje está formado por pares (x_t, d_t) , donde x_t es un vector con datos procedentes de la serie temporal y d_t es un escalar, con la propuesta de operación “óptima” para la sesión $t+1$, tras cerrarse la sesión t . Se puede considerar que d_t es el

valor que debería haber propuesto el indicador en t (p_t), para que pudiéramos considerar que ha realizado una propuesta correcta. Tanto d como p pueden tomar valores entre -1.0 y +1.0 ($-1.0 \leq d_t \leq +1.0$ y $-1.0 \leq p_t \leq +1.0, \forall t$). Las propuestas posibles son “vender al descubierto” o “posición corta” (Sell), “no operar” (NoOp) y “comprar” o “posición larga” (Buy). Los valores negativos próximos a -1.0 se interpretan como Sell; los valores positivos próximos a +1.0 se interpretan como Buy; y el resto de valores como NoOp. En la sección “Anexos. Cálculo de d ”, se describe como se han calculado los valores de d a partir de la serie temporal. El conjunto de aprendizaje, y otros conjuntos a los que nos referiremos más adelante, son conjuntos de pares, de la forma $C = \{(x_1, d_1), \dots, (x_t, d_t), \dots, (x_N, d_N)\}$, siendo N el número de sesiones del conjunto.

El entrenamiento o aprendizaje consiste en optimizar el modelo, conociendo los datos de entrada y salida. O sea, se trata de optimizar M_{i22} , en $d_t = M_{i22}(x_t)$, conociendo $C = \{(x_1, d_1), \dots, (x_t, d_t), \dots, (x_N, d_N)\}$. El objetivo es optimizar un modelo, de forma que sea capaz de generalizar y obtener propuestas para datos de entrada nuevos, es decir desconocidos para el sistema: obtener una propuesta $p_t = M_{i22}(x_t)$, para $x_t \notin C$.

El proceso de aprendizaje permite obtener una población de individuos, cada uno de los cuales es un indicador del tipo i22. Tras el proceso de aprendizaje tiene lugar un proceso de selección, en el que se trata de obtener, mediante un criterio objetivo, aquellos individuos que parecen aptos para ser un buen modelo del mercado, es decir, que parecen buenos candidatos para generalizar fuera del conjunto de aprendizaje.

2. Evaluación

2.1. Selecciones evaluadas

Se evaluaron tres modelos (LR, AGR14 y AGR16) para AAPL y GOOG.

2.2. Conjunto de evaluación

Los modelos seleccionados han sido evaluados sobre un conjunto de evaluación C_{EV} . Ningún elemento de este conjunto de evaluación había sido explorado previamente por los modelos. O sea, el conjunto C_{EV} no contiene elementos que existan en los conjuntos de aprendizaje o validación: $\forall (x_t, d_t) \in C_{EV}, (x_t, d_t) \notin C_{APR}, (x_t, d_t) \notin C_{VAL}$.

El conjunto de evaluación está formado, para los tres tickers del apartado anterior por la serie temporal diaria preprocesada, comprendida entre el día 8 de junio de 2020 el día 3 de junio de 2022, ambos incluidos: $C_{EV} = \{(x_t, d_t) | 2020 - 06 - 08 \leq t \leq 2020 - 06 - 03\}$.

2.3. Cálculo del beneficio obtenido sobre el conjunto de evaluación

La medida que se utilizará como resultado final de la evaluación, también en secciones posteriores, cuando se evalúen indicadores del Análisis Técnico, será el beneficio anualizado. Para calcularlo se procederá de la siguiente forma: Para cada sesión del conjunto o ventana de evaluación el indicador presenta una propuesta de operación. Se realiza la operación, si procede (ya que la propuesta puede ser “no operar”), y se calcula

el beneficio o pérdida obtenido. El resultado final es la suma de los beneficios (o pérdidas) de las operaciones individuales. A continuación se explica con más detalle.

Se considera un único ticker y un conjunto de sesiones consecutivas para ese ticker (el conjunto o ventana de evaluación). El número de elementos de la ventana es N .

En primer lugar, se obtiene una lista de todas las operaciones que el indicador propone para ese ticker y esa ventana (Ver “Anexos. Generación de operaciones”).

Se obtiene el beneficio (expresado en tanto por cien) de cada operación individual de la siguiente forma:

- 1) Si se trata de una operación “compra” (Buy): $r_{oper} = 100 * \ln\left(\frac{p_f}{p_i}\right) - costes$, siendo p_i el precio inicial (precio de compra), p_f el precio final (precio de venta) y $costes$, los costes asociados a la operación. Se ha tomado $costes = 0.2$ (0.2%)
- 2) Si se trata de una operación “venta al descubierto” (Sell): $r_{oper} = 100 * \ln\left(\frac{p_i}{p_f}\right) - costes$, siendo p_i el precio inicial (precio de venta), p_f el precio final (precio de compra) y $costes$, los costes asociados a la operación. Se ha tomado $costes = 0.2$ (0.2%)

El beneficio total, para todas las operaciones propuestas por el indicador para ese ticker sobre esa ventana ($O \equiv \{operaciones\}$), será: $r_{total} = \sum_{oper \in O} r_{oper}$.

Finalmente, el beneficio anualizado, más útil para realizar comparaciones, será: $r = \frac{22*12*r_{total}}{N}$.

Hay que notar que, tanto la forma de calcular el beneficio, como los algoritmos para generar operaciones, son los mismos para todos los indicadores evaluados. La única diferencia está en la acción que hemos denominado “obtenemos la propuesta para la siguiente vela” (ver anexo A.2): aquí es donde cada indicador proporciona un resultado ($p \in [-1.0, +1.0]$) diferente, obtenido de su propio generador de propuestas. Es decir “obtenemos la propuesta para la siguiente vela” significa, en el caso de i22, $p_t = M_{i22}(x_t)$. Para los indicadores del AT utilizados se describirá, más adelante, la generación de propuestas.

2.4. Resultados

En la siguiente tabla se presentan los resultados. Los valores de la tabla son el beneficio (logreturn) anualizado obtenido por las operaciones realizadas a lo largo del periodo descrito en 2.2.

Ticker	LR	AGR14	AGR16
AAPL	49.32	48.83	45.65
GOOG	83.39	56.66	55.28
Media	66.36	52.75	50.47

En el Anexo B se describen algunos archivos con información adicional sobre el proceso de evaluación.

3. Indicadores del Análisis Técnico

Se ha realizado la evaluación de algunos indicadores del Análisis Técnico sobre el mismo conjunto de evaluación (C_{EV}). Los indicadores evaluados han sido MACD, Estocástico, Demand Index (DI) y una combinación de los tres, que llamaremos MIX.

Se han utilizado los mismos algoritmos para realizar automáticamente operaciones durante todo el periodo de tiempo correspondiente al conjunto de evaluación y para calcular el beneficio obtenido. La única diferencia tiene que ver con el algoritmo “opera”, y más concretamente con la sentencia que dice “obtenemos la propuesta para la siguiente vela”. En este caso, la forma de obtener las propuestas se describe en el anexo C.

En la siguiente tabla se presentan los resultados obtenidos para los indicadores mencionados. Los valores de la tabla son el beneficio (logreturn) anualizado obtenido por las operaciones realizadas a lo largo del periodo descrito en 2.2.

Ticker	MACD	Estocástico	DI	MIX
AAPL	24.82	-30.54	0.39	19.22
GOOG	-0.26	33.33	13.99	-8.67
Media	12.28	1.40	7.19	5.28

ANEXOS**A1. Cálculo de d**

Algoritmo para calcular d en los pares (\vec{x}, d)

El valor d es la, teóricamente, mejor propuesta del modelo en t para un conjunto de variables de entrada \vec{x} . En nuestro caso se ha de cumplir que $-1.0 \leq d \leq +1.0$.

begin

$PMAX = 3$

$GOPT = 3$

$d = 0.0$

$p_{ini} = p_{open}(t)$

$i = 0$

while($i < 6$):

if $\left(p_{high}(t + i) > p_{ini} * \left(1 + \frac{PMAX}{100}\right)\right)$ *and* $p_{low}(t + i) < p_{ini} * \left(1 - \frac{PMAX}{100}\right)$:

“dejamos lo que hubiera por la alta volatilidad”

exit while

else:

if $(p_{close}(t + i) > p_{ini})$:

“el precio sube”

if $\left(d < -\frac{1}{3}\right)$:

“no cambiamos el valor de d porque lo consideramos un Sell consolidado”

else:

$$d' = \frac{p_{close}(t+i) - p_{ini}}{p_{ini}} * \frac{100}{GOPT}$$

if $(d' > 1.0)$: $d' = 1.0$

if $(d' > d)$: $d = d'$

elseif $(p_{close}(t + i) < p_{ini})$:

“el precio baja”

if $\left(d > \frac{1}{3}\right)$:

“no cambiamos el valor de d porque lo consideramos un Buy consolidado”

else:

$$d' = \frac{p_{close}(t+i) - p_{ini}}{p_{ini}} * \frac{100}{GOPT}$$

$$if(d' < -1.0): d' = -1.0$$

$$if(d' < d): d = d'$$

$$if(d = 1.0 \text{ or } d = -1.0):$$

“terminamos porque este resultado ya es inamovible”

exit while

$$i = i + 1$$

return d

end

COMENTARIOS:

- PMAX es la pérdida máxima. El valor que usamos, es el 3% (PMAX=3). Por eso, si los precios máximo (high) o mínimo (low) sobrepasan el precio inicial en ese porcentaje, dejamos lo que hubiera en d, ya que la alta volatilidad hace imposible afinar más el valor de d.
- El valor de d se puede considerar como el grado de aproximación del beneficio obtenido con respecto al beneficio óptimo esperado (GOPT) de una operación individual. En la práctica usamos un beneficio óptimo del 3% (GOPT=3). Por eso un beneficio del 3% o superior corresponde a un valor d = +1.0 (si es Buy) o d = -1.0 (si es Sell). Por ejemplo, un beneficio del 2% daría $d = \frac{2}{100} * \frac{100}{3} = \frac{2}{3}$, y un beneficio del 1% daría $d = \frac{1}{100} * \frac{100}{3} = \frac{1}{3}$.
- Con $d = \pm \frac{1}{3}$ (beneficio del 1%) consideramos que ya no se puede cambiar el signo de la operación. El “stop loss” se habrá movido en el sentido del beneficio, consolidando el beneficio obtenido. El valor de d puede aumentar (en términos absolutos) pero no cambiar de signo.

A2. Generación de operaciones

Para calcular el beneficio obtenido durante un periodo (o, de forma equivalente, sobre un conjunto de sesiones) por un indicador, hay que obtener la lista de operaciones propuestas por ese indicador durante el periodo en cuestión.

Cada operación generada es una entidad con un conjunto de propiedades: tipo (Buy o Sell), precio de entrada o inicio, precio de salida o fin, stop-loss y logreturn. Estas propiedades se identifican como *oper.tipo*, *oper.pi*, *oper.pf*, *oper.sl* y *oper.logreturn*

Como valor inicial para stop-loss se toma el siguiente:

- para operaciones Buy, $sl = Precio_{ini} * (1 - SL/100)$
- para operaciones Sell, $sl = Precio_{ini} * (1 + SL/100)$

A continuación se describen los algoritmos que obtienen la lista de operaciones propuestas para un determinado conjunto de sesiones.

Algoritmo opera

begin opera

“nos posicionamos en la primera vela de la ventana”

“obtenemos la propuesta para la siguiente vela”

“nos movemos a la siguiente vela”

while(hay velas):

if(condicionesParaOperar(Buy, propuesta)):

“generar nueva operación Buy”

“guardar en la lista de operaciones”

“guardar en la lista de operaciones Buy abiertas”

elseif(condicionesParaOperar(Sell, propuesta)):

“generar nueva operación Sell”

“guardar en la lista de operaciones”

“guardar en la lista de operaciones Sell abiertas”

if("hay elementos en la lista de operaciones Buy abiertas"):

comprobarCierreBuy

if("hay elementos en la lista de operaciones Sell abiertas"):

comprobarCierreSell

“obtenemos la propuesta para la siguiente vela”

“nos movemos a la siguiente vela”

while("quedan operaciones abiertas"):

if("hay elementos en la lista de operaciones Buy abiertas"):

comprobarCierreBuy

if("hay elementos en la lista de operaciones Sell abiertas"):


```

        comprobarCierreSell
    if("aún queda alguna operación abierta):
        cierreIncondicional
    return "lista de operaciones"
end

```

Algoritmo condicionesParaOperar

Este algoritmo comprueba si se dan las condiciones para realizar una operación. Devuelve True, si se recomienda operar, o False, si no se recomienda operar.

Entradas: *acc* indica el tipo de operación (Buy o Sell) que se consulta; *propuesta* es la propuesta del indicador para la sesión actual

```

begin condicionesParaOperar(acc, propuesta)
    "la operación se puede rechazar (False) por haberse sobrepasado el número total de
    operaciones abiertas (NUM_MAX), o bien por haberse sobrepasado el número total de
    operaciones Buy (NUM_MAX_BUY) o Sell (NUM_MAX_SELL) abiertas"
    if("no hay operaciones abiertas"):
        if(acc = Sell and propuesta > -COEF_MIN_1):
            return False
        elseif(acc = Buy and propuesta < COEF_MIN_1):
            return False
        else:
            return True
    else:
        "segunda o tercera operación"
        if(acc = Sell and propuesta > -COEF_MIN_2):
            return False
        elseif(acc = Buy and propuesta < COEF_MIN_2):
            return False
        else:
            return True
end

```

end

COMENTARIOS:

- NUM_MAX es el número máximo de operaciones abiertas. El valor que usamos es NUM_MAX=2. Las operaciones abiertas simultáneamente, en un momento de terminado, pueden ser dos Buy, dos Sell, o una Buy y otra Sell. Los parámetros NUM_MAX_BUY y NUM_MAX_SELL, por lo tanto, se consideran también 2.
- COEF_MIN_1 es el valor absoluto mínimo de la propuesta para iniciar una operación (Buy si es positivo, Sell si es negativo), siempre que no haya ya una operación abierta. Se ha usado COEF_MIN_1=0.80
- COEF_MIN_2 es el valor absoluto mínimo de la propuesta para iniciar una operación (Buy si es positivo, Sell si es negativo) cuando ya hay alguna operación abierta. Se ha usado COEF_MIN_2=0.90

Algoritmo *comprobarCierreBuy*

Este algoritmo comprueba si hay que cerrar operaciones Buy que están abiertas.

begin comprobarCierreBuy

“obtener la primera operación Buy que aún está abierta”

while("hay operaciones abiertas"):

if ($Precio_{close}(t) \geq Precio_{open}(t)$):

“se trata de una vela blanca (alcista), o un doji”

if ($Precio_{Low} \leq oper.sl$):

cerrarOperacion

elseif ($Precio_{High} \geq oper.pi * (1 + TS/100)$ and $oper.sl < Precio_{High} * (1 - TS/100)$):

$oper.sl = Precio_{High} * (1 - TS/100)$

if ($Precio_{close} \leq oper.sl$):

cerrarOperacion

else:

“se trata de una vela negra (bajista)”

if ($Precio_{High} \geq oper.pi * (1 + TS/100)$ and $oper.sl < Precio_{High} * (1 - TS/100)$):

$oper.sl = Precio_{High} * (1 - TS/100)$

if ($Precio_{Low} \leq oper.sl$):

cerrarOperacion

end

Algoritmo comprobarCierreSell

Este algoritmo comprueba si hay que cerrar operaciones Sell que están abiertas.

begin comprobarCierreSell

“obtener la primera operación Sell que aún está abierta”

while("hay operaciones abiertas"):

if ($Precio_{Close}(t) \leq Precio_{Open}(t)$):

“se trata de una vela negra (bajista), o un doji”

if ($Precio_{High} \geq oper.sl$):

cerrarOperacion

elseif ($Precio_{Low} \leq oper.pi * (1 - TS/100)$ and $oper.sl > Precio_{Low} * (1 + TS/100)$):

$oper.sl = Precio_{Low} * (1 + TS/100)$

if ($Precio_{Close} \geq oper.sl$):

cerrarOperacion

else:

“se trata de una vela blanca (alcista)”

if ($Precio_{Low} \leq oper.pi * (1 - TS/100)$ and $oper.sl > Precio_{Low} * (1 + TS/100)$):

$oper.sl = Precio_{Low} * (1 + TS/100)$

if ($Precio_{High} \geq oper.sl$):

cerrarOperacion

end

Algoritmo cerrarOperacion

Cerrar una operación consiste en modificar el valor de algunas de sus propiedades:

- La propiedad estado toma el valor “Close” (*estado* = "Close")
- El precio final toma el valor stop-loss (*pf* = *sl*)
- También se guarda la fecha en que se cerró la operación.

Algoritmo *cierreIncondicional*

Este algoritmo cierra todas las operaciones que quedan abiertas. El precio de cierre es el precio último disponible. Se invoca cuando se ha llegado al final de la ventana (ya no quedan más sesiones) y aún hay operaciones abiertas.

Por lo tanto, hace lo mismo que *cerrarOperacion*, excepto que, en este caso, $pf = Precio_{close}$.

- La propiedad estado toma el valor "Close" ($estado = "Close"$)
- El precio final toma el valor del último precio de cierre disponible ($pf = Precio_{close}$)
- También se guarda la fecha en que se cerró la operación.

B. Archivos

B.1. Archivos oper_tttt_D_id.txt

En el nombre del archivo, “tttt” se sustituye por el código del ticker, e “id” es el identificador del modelo.

Estos archivos contienen la lista de todas las operaciones realizadas durante la evaluación. Se indica el tipo de operación, la fecha de inicio y la fecha de fin, el precio de entrada y el precio de salida, y el beneficio (logreturn) obtenido. Tras la lista de operaciones se indica el beneficio total anualizado.

A continuación, hay una lista de propuestas, en la que aparecen, para todas las sesiones del conjunto de evaluación, la propuesta del indicador para la siguiente sesión.

B.2. Archivos plot_tttt_D_id.txt

Como en el caso anterior, en el nombre del archivo, “tttt” se sustituye por el código del ticker, e “id” es el identificador del modelo.

Estos archivos contienen, para cada sesión del conjunto de evaluación, la siguiente información: fecha de la sesión, propuesta de i22 (<i22>), las medias móviles simples de las propuestas, de 3, 5, 10 y 20 sesiones (<i22SMA3>, <i22SMA5>, <i22SMA10> y <i22SMA20>), y las medias móviles exponenciales de las propuestas, de 3, 5, 10 y 20 sesiones (<i22EMA3>, <i22EMA5>, <i22EMA10> y <i22EMA20>).

El formato del archivo permite la presentación gráfica de la información, desde una plataforma de trading.

C. Algoritmos para generar propuestas en MACD, Estocástico, DI y MIX**C.1. MACD**

if($MACD(t) > Signal(t)$ and $PMACD(t) > 0.10$):

propuesta = *Buy*

elseif($MACD(t) < Signal(t)$ and $PMACD(t) < -0.10$):

propuesta = *Sell*

else:

propuesta = *NoOp*

PMACD se calcula de la siguiente forma:

$$PMACD(t) = \frac{PMACD(t) - PMACD(t - 1)}{|PMACD(t - 1)|}$$

C.2. Estocástico

if($K(t) > D(t)$ and $K(t) < 80$):

propuesta = *Buy*

elseif($K(t) < D(t)$ and $K(t) > 20$):

propuesta = *Sell*

else:

propuesta = *NoOp*

Siendo:

$$K(t) = \frac{(\text{Precio}_{\text{close}} - \text{Precio}_{\text{min}}) * 100}{\text{Precio}_{\text{max}} - \text{Precio}_{\text{min}}}$$

$\text{Precio}_{\text{max}}$ = Máximo $\text{Precio}_{\text{close}}$ de las últimas 14 sesiones

$\text{Precio}_{\text{min}}$ = Mínimo $\text{Precio}_{\text{close}}$ de las últimas 14 sesiones

$D(t)$ = Media móvil simple de las últimas 3 sesiones

C.3. Demand Index

if($SMA(DI, 2) > 0$ and $dSMA(DI, 2) > 0$):

```

    propuesta = Buy
elseif( $SMA(DI, 2) < 0$  and  $dSMA(DI, 2) < 0$ ):
    propuesta = Sell
else:
    propuesta = NoOp

```

Siendo:

$SMA(DI, 2)$, la media móvil simple de dos sesiones del DI.

$dSMA(DI, 2)$, la pendiente de la media móvil de dos sesiones del DI.

C.4. MIX

"obtener las propuestas de MACD, Estocástico y DI"

"calcular N_{Buy} = número de propuestas Buy"

"calcular N_{Sell} = número de propuestas Sell"

```

if( $N_{Buy} > 1$  and  $N_{Sell} = 0$ ):
    propuesta = Buy
elseif( $N_{Sell} > 1$  and  $N_{Buy} = 0$ ):
    propuesta = Sell
else:
    propuesta = NoOp

```