Universidad Rey Juan Carlos
Facultad de Ciencias de la Economía y la Empresa
Universidad Rey Juan Carlos

Aplicando inteligencia artificial a un sistema de trading.

Resumen:

En este trabajo se aborda la aplicación de la inteligencia artificial a la predicción bursátil. Se desarrollan 2 ideas principales destinadas a optimizar una estrategia de trading mediante el uso de potentes algoritmos escritos en el lenguaje de programación Python.

TRABAJO FIN DE GRADO

Apellidos y nombre: Aritz Lizaso Lopez-Mora

D.N.I.: 53765539a

Correo electrónico: aritz.lizaso@alumnos.urjc.es

Director: Piedad Olmos Rodríguez

Grado en Administración y dirección de empresas

Grupo: Campus Móstoles

Curso: 2023/2024 – convocatoria: Enero

Contenido

Introducción

En los últimos años se ha dado un gran avance en el campo del “machine learning” debido a un aumento en la capacidad de computación que ha permitido mejorar considerablemente los tiempos de entrenamiento que este tipo de algoritmos requieren.

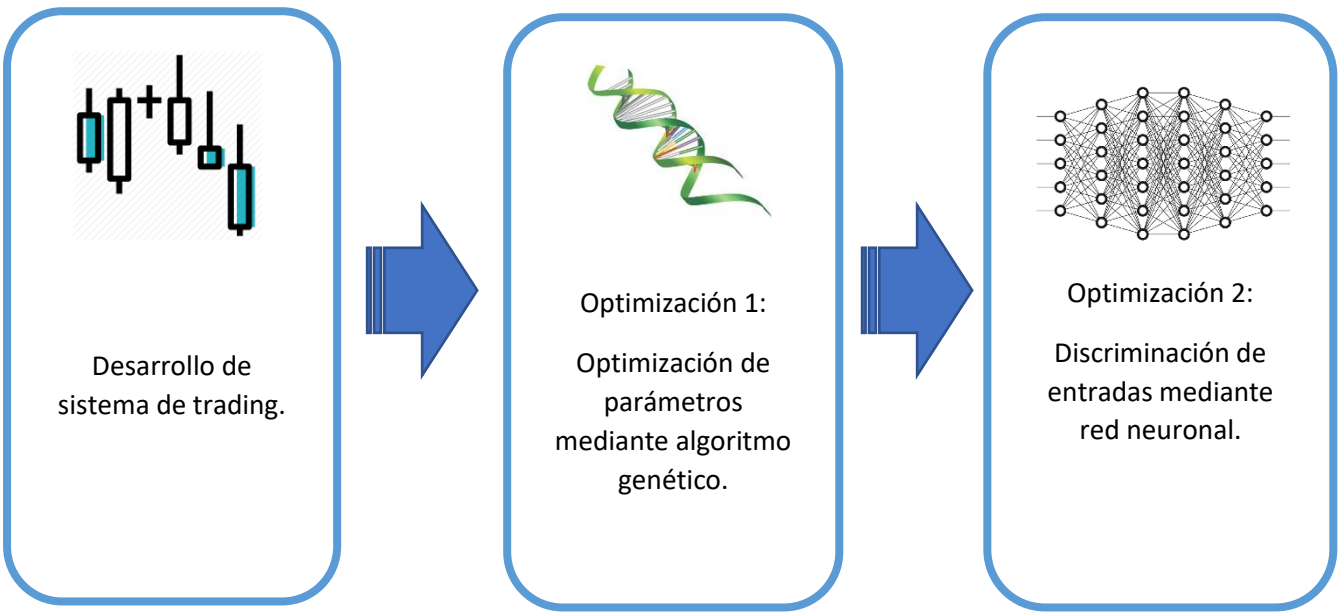
Esto es una oportunidad que desde el campo de los mercados financieros no se ha desaprovechado y es por eso que han aparecido diversos fondos cuantitativos que tratan de solventar los problemas de predicción de valor de los distintos activos ayudándose del uso de algoritmos de “machine learning”.

En este trabajo de fin de grado se utilizarán diversas técnicas basadas en inteligencia artificial para tratar de potenciar los conocimientos adquiridos a lo largo del grado en predicción de mercados financieros.

Objetivo

El objetivo de este trabajo será desarrollar una estrategia de "trading" en el mercado de divisas Forex con unas reglas determinadas, evaluar dicha estrategia y a continuación pasar a optimizar su rentabilidad, para ello utilizaremos:

- En una primera fase un algoritmo genético con el objetivo de optimizar los parámetros de dicha estrategia.
- En una segunda fase se tratará de entrenar una red neuronal cuyo objetivo sea discriminar las entradas de dicho algoritmo, eliminando así aquellas que no vayan a resultar exitosas por el contexto del mercado.



Desarrollo de sistema de trading

El primer paso de este trabajo será desarrollar un sistema de trading. Para esta tarea se ha elegido el mercado de divisas FOREX. Se ha elegido este mercado por distintos motivos que facilitan la realización de este estudio:

- Facilidad de acceso a datos
- Permite operar de forma apalancada
- Bajas comisiones
- Permite entradas en largo y en corto (Apostar por una revalorización del activo o por una depreciación)
- Gran liquidez

En este mercado se negocian pares de divisas, que no son más que los distintos tipos de cambio de unas divisas a otras.

Se diseñará una estrategia capaz de aprovechar movimientos de reversión a la media dentro de la tendencia dominante. Para ello será necesario detectar tanto la tendencia dominante como los estados de sobrecompra y sobreventa.

La estrategia se apoyará en tres indicadores de análisis técnico para dicho cometido:

- Media móvil exponencial de corto periodo
- Media móvil exponencial de largo periodo
- Williams % R

Las medias móviles exponenciales calculan el precio medio de n periodos atrás dando mayor peso a los periodos recientes.

$$EMAt = \alpha * precio\ actual + (1- \alpha) * EMA\ t\text{-}1$$

Donde:

α es una constante suavizada con un valor entre 0 y 1. La fórmula de ***α*** es:

$$2 / (cantidad\ de\ períodos\ seleccionados + 1)$$

EMA t-1 se trata de la EMA calculada en el periodo anterior.

Utilizaremos ambas EMAs para saber si nos encontramos en tendencia alcista (EMA de corto periodo superior a EMA de largo periodo) o de lo contrario nos encontramos en tendencia bajista (EMA de corto periodo inferior a EMA de largo periodo)

- Zona azul: Tendencia alcista
- Zona Roja: Tendencia bajista

El indicador williams percent Range es un indicador dinámico que determina el estado de sobrecompra/sobreventa.

$$Williams\ \%\ R = (Nth\ High - Close) / (Nth\ High - Nth\ Low) \times (-100)$$

Donde:

Nth High: Precio máximo del periodo analizado.

Nth Low: Precio mínimo del periodo analizado

Close: El precio de cierre del periodo analizado

Este indicador se construye con el objetivo de cuantificar la distancia a la que está el mercado de los limites de un rango reciente. Un valor de -100% indica que el precio actual está en el mínimo más bajo de los n periodos en los que se calcula, mientras que un valor de 0% indica justo lo contrario.

Con estos indicadores la estrategia tendrá información sobre la tendencia del mercado (EMAs) y también sobre potenciales puntos de reversión a la media (Williams %R).

Selección de activos

En el mercado Forex el universo de activos para operar (pares de divisas) es muy limitado, y más aún si lo reducimos a los activos con un gran volumen de negociación.

Por esta razón el algoritmo se encargará de negociar de forma individual el par deseado.

En este trabajo se va a desarrollar un método de optimización aplicable a cada par de forma individualizada.

Por esta razón nuestro algoritmo no tendrá que hacer frente a la decisión de selección de activos ya que será el gestor el encargado de decidir a que activos desea aplicar esta estrategia y su correspondiente optimización.

Asignación de capital

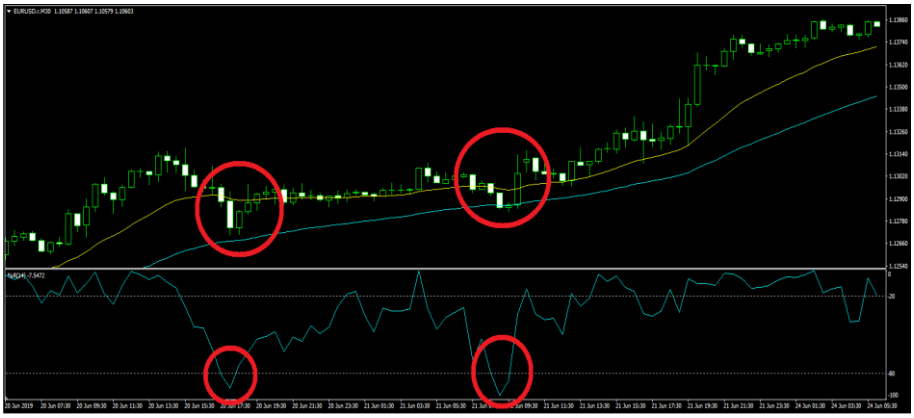
La asignación de capital vendrá dada por el apalancamiento que el gestor decida utilizar en su operativa. Las operaciones tendrán siempre el mismo volumen de capital siendo más o menos rentables en función del riesgo que se desee asumir. En nuestra optimización utilizaremos por defecto un apalancamiento de 1:3, un nivel de apalancamiento razonable y común en este mercado. El capital inicial es 100.000 USD por defecto.

Señales de entrada a mercado

La estrategia diseñada tendrá dos tipos de señales de entrada: una para entradas en largo y otra para entradas en corto.

- **Entrada en largo**: El algoritmo realizara una entrada en largo siempre que se den las siguientes dos condiciones:
    - La EMA de corto periodo es mayor que la EMA de largo periodo
    - El indicador Williams % R marcará sobreventa
- **Entrada en corto**: El algoritmo realizará una entrada en corto siempre que se den las siguientes condiciones:
    - La EMA de corto periodo es menor que la EMA de largo periodo
    - El indicador Williams % R marcará sobrecompra

Lo que la estrategia busca es aprovechar correcciones al alza mientras el mercado mantiene una tendencia alcista y siempre que se hayan producido correcciones a la baja y viceversa. A continuación, se muestra una imagen con ejemplos de señales de entrada en largo:



Señales de salida

Las órdenes que lance el algoritmo irán acompañadas de una orden stop loss para detener las perdidas en caso de que el precio se mueva en la dirección opuesta a la deseada y una orden take profit para recoger las ganancias una vez que el movimiento deseado se haya producido.

Esta estrategia no realizará cierres parciales de la posición lo que significa que siempre se cerrará el total de la orden.

La distancia a la que se colocaran estas órdenes vendrá definida por un parámetro de la propia estrategia.

Gestión de la cartera

- **Capital disponible al realizar la venta:** En el mercado Forex no se recibe dinero cuando se realiza una operación en corto.
- **En el caso de que no haya capital disponible suficiente:** Si no hay capital disponible al algoritmo no se le permite realizar operaciones.
- **Límites de concentración de riesgo:** El riesgo quedará limitado por el apalancamiento deseado y la distancia de la orden stop loss.
- **Comisiones de compra-venta:** Dependerá del bróker utilizado. Por defecto el algoritmo calculará una comisión única por trade realizado de 5 unidades monetarias por lote negociado. (En Forex un lote equivale a 100.000 unidades de la divisa de cotización)
- **Umbral de eficiencia:** La estrategia no requiere de ningún umbral de eficiencia ya que es escalable por el apalancamiento.

- **Umbral de arrastre:** En Forex el volumen de negociación es demasiado grande como para que un solo inversor pueda mover el mercado por muy grande que sea, por lo que no existe umbral de arrastre.
- **Tipo de órdenes:** Se introducirá al mercado siempre órdenes de mercado. Siempre se supone ejecución debido al gran volumen negociado en los grandes pares de Forex.
- **Reinversión de los beneficios:** Por defecto el algoritmo no reinvertirá los beneficios.
- **Ventana de datos:** La ventana de datos utilizada será independiente, teniendo cada indicador la suya propia y estática por defecto.
- **Tipo de stop loss:** El stop loss siempre se respetará, no se cerrarán operaciones antes de que toquen la orden stop loss o la orden take profit.
- **Criterio de quiebra:** Perder todo el capital inicial.
- **Gestión de la tesorería:** En Forex no es necesario reservar dinero para pagar las comisiones, la comisión se descuenta del beneficio de la operación en el momento de entrar a mercado.
- **Gestión del riesgo de divisa:** Nuestro algoritmo por defecto solo negocia pares cotizados en dólares estadounidenses, por lo que no es necesario cubrirse del riesgo de divisa. En caso de ser necesario se asumirá.

Datos

Los datos utilizados para el desarrollo de este algoritmo han sido extraídos del software Tic Data Suite, concretamente de la fuente de TrueFX. Todos los datos usados se encuentran en la carpeta “datos” del archivo de entrega.

Los datos utilizados para el desarrollo del backtest de la estrategia serán de periodos de 30 minutos. Se utilizará este timeframe debido a que se trata de una estrategia intradía y por ello requiere un timeframe menor al diario, y a su vez no queremos que

el estudio de la idea sea muy pesado ya que para la optimización mediante el algoritmo genético necesitaremos realizar un backtest lo más eficiente y rápido posible.

La homogeneización la realiza automáticamente la plataforma Tic Data Manager incluida en el software Tic Data Suite.

Al tratarse del mercado Forex no será necesario la realización de ajustes por Split, contrasplit o dividendos.

Backtesting

Para desarrollar el backtest de la estrategia se ha utilizado como base la librería backtrader. Esta decisión se ha tomado debido a las numerosas características ya programadas que se pueden aprovechar y sobre todo por un mayor nivel de eficiencia y velocidad en la ejecución del backtest.

El código base de la estrategia y su respectivo backtest se encuentran en la carpeta "código" del archivo de entrega.

Una vez en el código se han desarrollado los siguientes pasos:

Datos**:** Se ha procedido a la carga de los datos y su preparación para la librería backtrader. Para la elaboración del backtest solo se han tenido en cuenta los datos desde 2014 hasta 2018, dejando los datos del presente 2019 para validar la robustez del modelo proporcionado por la primera optimización mediante un algoritmo genético. Además del EURUSD se han cargado los datos de los cuatro pares que utilizaran en esta estrategia.

Estrategia**:** Se ha definido la estrategia anteriormente descrita adaptada al funcionamiento de la librería backtrader. También se han programado las funciones "notify_order" y "notify_trade" que se encargaran de mostrar de forma visual todo lo que va ocurriendo a lo largo de la ejecución del backtest.

Backtesting**:** Se ha ejecutado el backtest siguiendo las reglas de backtrader. El par a negociar será el EURUSD por defecto, que es el par que se utilizará como ejemplo de la optimización de este trabajo. Se han añadido "Observers" y "Analyzers", que son funciones de la librería backtrader que permiten recoger información adicional para obtener ciertas características en los gráficos y en las estadísticas del informe.

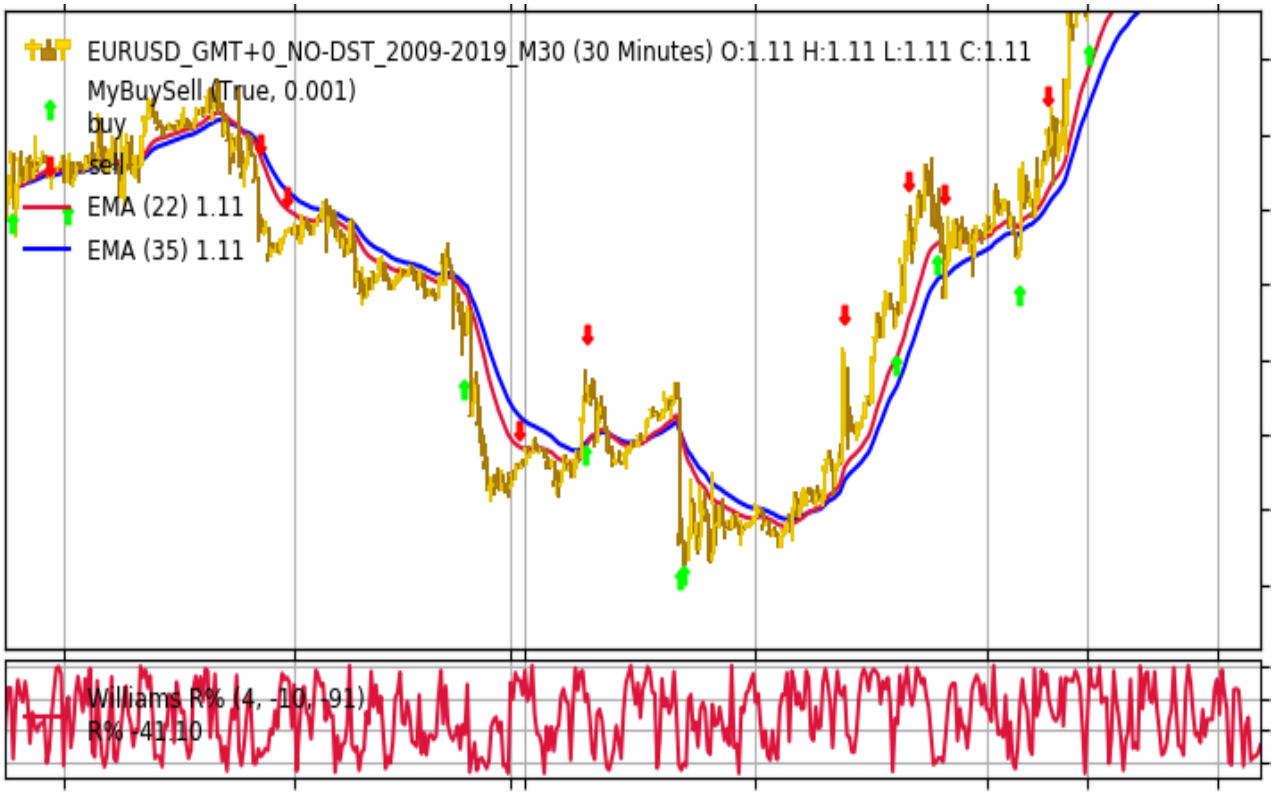
Representación visual**:** Para la representación visual de los datos se ha creado un gráfico interactivo por el que se puede navegar a cualquier periodo deseado y que permite ser dividido en distintos gráficos que muestren diferentes periodos de tiempo para una mejor visualización. Este gráfico general se divide en tres subgráficos que contienen:

- Valor de la cartera: Un gráfico lineal que muestra la evolución del valor del capital a lo largo del periodo en el que se ha realizado el backtest.
- Beneficio de cada operación: Gráfico de puntos en el que cada punto hace referencia a una operación distinta. Será verde si la operación ha tenido beneficios y rojo si la operación ha tenido pérdidas. El eje Y nos mostrará la cuantía de esos beneficios o pérdidas.
- Entradas y salidas del algoritmo: Gráfico de velas japonesas de periodos de treinta minutos en el que se muestran los tres indicadores técnicos utilizados y todas las entradas y salidas que realiza el algoritmo. En este gráfico podemos comprobar la actividad de el algoritmo en el mercado.

A continuación, se muestra un ejemplo del gráfico general a lo largo de varios meses y también una ampliación del gráfico de entradas en el que se pueda apreciar la actividad.

Varios meses

Ampliación de varios días



Reporte**:** Para la realización del reporte se han recolectado diversas estadísticas mediante la ejecución del backtest de cara a poder resumir de una forma clara y precisa la actividad del algoritmo. Se ha creado un primer reporte rápido que se muestra en el notebook en el que solo se incluyen las estadísticas debido a que el notebook ya posee un gráfico interactivo. También se ha creado la opción de generar un reporte en formato PDF que se guarda en el directorio de trabajo y que contiene tanto las estadísticas recopiladas como el gráfico general del periodo en el que se ha ejecutado el backtest.

A continuación, se muestra un ejemplo del reporte que se podrá encontrar dentro del notebook.

Ejemplo de reporte dentro de notebook:

```
                    *** BACKTEST REPORT ***
_____________________________________________________________________

------------------- Strategy parameters --------------------
EMA Fast: 22
EMA Slow: 35
Williams %R period: 4
Williams %R upperband: -10
Williams %R lowerband: -91
Stop loss: 169
Take profit: 363
Lot size: 3

---------------------- General Results -----------------------

Initial portfolio value: 100000.00
Final portfolio value: 114880.66
Total gross profit: 17937.00
Total netprofit: 14873.96
Commisions: 3063.04
Total return: 14.88%
Expected payoff: 734.25

------------------------- Drawdown --------------------------

Drawdown: 2.63%              Absolute Drawdown: 3107.54
Max drawdown: 6.47%          Max absolute drawdown: 6692.34

-------------------------- Trades ---------------------------

Total trades: 182
Long trades: 86 (0.47%)         Short trades: 95 (0.52%)
Profit trades: 69 (0.38%)       Loss trades: 112 (0.62%)
Consecutive wins: 6             Consecutive losses: 7
Average profit trade: 1076.15   Average loss trade: -530.18
Max win: 1405.32                Max loss: -964.7
```

Pasamos a realizar el backtesting de la estrategia con los parámetros por defecto:

- _ema_fast=22
- _ema_slow=35
- _williams_period=4
- _williams_upperband=-10
- _williams_lowerband=-91
- _stoploss=169
- _takeprofit=363
- _lot = 3

Los resultados obtenidos al realizar el backtesting desde 01-01-2014 hasta 31-12-2018 son los siguientes:

```
                    *** BACKTEST REPORT ***
_____________________________________________________________________

------------------- Strategy parameters --------------------
EMA Fast: 19
EMA Slow: 50
Williams %R period: 14
Williams %R upperband: -20
Williams %R lowerband: -80
Stop loss: 150
Take profit: 150
Lot size: 3

--------------------- General Results ----------------------

Initial portfolio value: 100000.00
Final portfolio value: 86459.60
Total gross profit: 38751.00
Total netprofit: -13882.83
Commisions: 52633.83
Total return: -13.54%
Expected payoff: 478.98

------------------------- Drawdown --------------------------

Drawdown: 25.4%               Absolute Drawdown: 29439.74
Max drawdown: 33.29%          Max absolute drawdown: 38580.3

------------------------- Trades --------------------------
```

```
Total trades: 3019
Long trades: 1516 (0.5%)          Short trades: 1502 (0.5%)
Profit trades: 1581 (0.52%)       Loss trades: 1437 (0.48%)
Consecutive wins: 9               Consecutive losses: 12
Average profit trade: 452.93      Average loss trade: -507.98
Max win: 3976.59                  Max loss: -4996.19
```

El informe completo se encuentra en la carpeta “reportes” del archivo de entrega.

Como se puede observar la estrategia inicial obtiene pérdidas durante este periodo (-13882.83 USD). También se puede observar que la estrategia sin comisiones nos da un resultado positivo (38751.00USD).

Es probable que optimizando sus parámetros se pueda conseguir que esta estrategia sea rentable y capaz de asumir sus comisiones. Para ello pasaremos al desarrollo de un algoritmo genético que nos ayude a solventar este problema de optimización.

Optimización 1: Algoritmo genético

Definición del proceso

Nos encontramos ante un problema de optimización de 7 parámetros que pueden tomar un amplio rango de valores cada uno. La principal dificultad es el casi infinito número de combinaciones de parámetros posibles que hay que evaluar como posibles soluciones.

Para el problema anteriormente planteado desarrollaremos un algoritmo genético, que resolverá la búsqueda de la combinación de parámetros de una forma mucho más eficiente. Este tipo de algoritmos realizan búsquedas aleatorias dentro de todo el

espacio posible y tratan de escalar a máximos locales ante la imposibilidad de evaluar todas las combinaciones.

En el algoritmo genético diseñado cada solución (combinación de parámetros a optimizar) será un cromosoma, y cada parámetro un gen. Un cromosoma estará formado por siete genes, cada parámetro a optimizar en la estrategia: 'ema_fast', 'ema_slow', 'williams_period', 'williams_upperband', 'williams_lowerband', 'stoploss', 'takeprofit'.

Para generar aleatoriamente cada parámetro se establecerá un rango con máximo y mínimo para cada uno de ellos:

- maxi = [35,65,30,-5,-65,250,500]
- mini = [5,35,3,-35,-95,50,50]

Estableceremos una población de doce cromosomas por cada generación de los cuales cuatro de ellos serán utilizados para engendrar las siguientes generaciones.

El procedimiento que seguirá este algoritmo será el siguiente:

1. Se creará una generación inicial aleatoria comprendida entre los rangos anteriormente descritos.
2. Se evaluará cada cromosoma mediante una función fitness que evaluará su puntuación mediante la realización de un backtesting. La puntuación estará basada en la rentabilidad obtenida a lo largo del periodo.
3. Se seleccionará los cuatro mejores cromosomas que serán encargados de engendrar la siguiente generación.
4. Mediante un proceso de “crossover” se recombinarán estos cromosomas hasta generar una población de ocho cromosomas más, que unidos a los cuatro padres formarán la nueva generación.
5. Para no perder la diversidad genética se procederá a un proceso de mutación a los ocho cromosomas generados anteriormente.

6. Volviendo al paso 2 repetiremos todo el procedimiento hasta que el algoritmo haya evaluado **n** números de generaciones y no haya detectado mejoras en las **x** últimas generaciones.

El algoritmo genético se ejecutará en el mismo notebook que el backtesting. Para no masificar el notebook se han definido todas las funciones que utilizara este algoritmo genético en un archivo ".py".

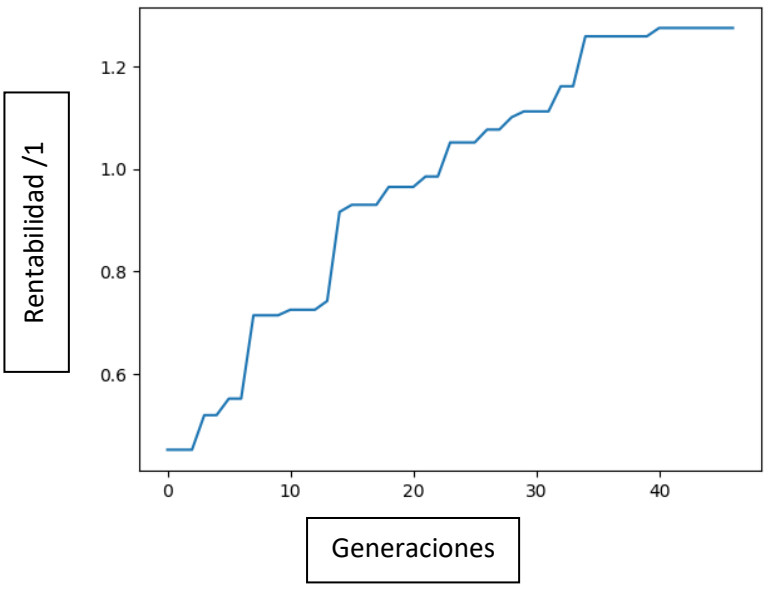
Todo el código estará contenido en la carpeta "código" del archivo de entrega.

Optimizando la estrategia

Para llevar a cabo esta optimización solo se utilizarán los datos de los cinco años anteriores, en este caso desde el año 2014 hasta el año 2018 (ambos incluidos), reservando los datos hasta el presente 30-08-2019 para validar la robustez del modelo proporcionado por el algoritmo genético.

Una vez ejecutada la optimización mediante el algoritmo genético los resultados son los siguientes:

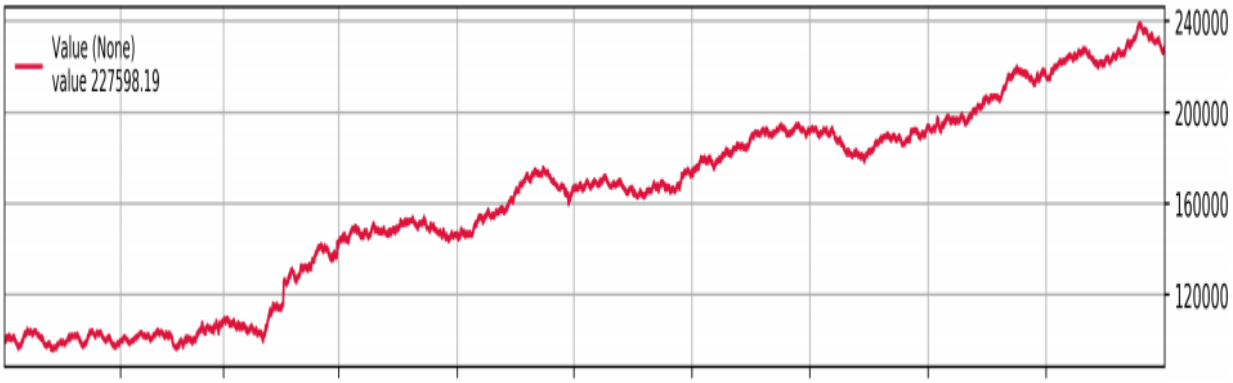
Evolución de la optimización



A lo largo de más de cuarenta generaciones el algoritmo genético ha optimizado la estrategia por encima del 120% de rentabilidad durante los cinco años.

```
                    *** BACKTEST REPORT ***
_____________________________________________________________

------------------- Strategy parameters --------------------
EMA Fast: 22
EMA Slow: 35
Williams %R period: 4
Williams %R upperband: -10
Williams %R lowerband: -91
Stop loss: 169
Take profit: 363
Lot size: 3

--------------------- General Results ----------------------

Initial portfolio value: 100000.00
Final portfolio value: 227598.19
Total gross profit: 166320.00
Total netprofit: 127306.76
Commisions: 39013.24
Total return: 127.6%
Expected payoff: 743.99

------------------------- Drawdown --------------------------

Drawdown: 4.79%              Absolute Drawdown: 11442.22
Max drawdown: 9.2%          Max absolute drawdown: 16620.68

------------------------- Trades ---------------------------

Total trades: 2244
Long trades: 1125 (0.5%)         Short trades: 1118 (0.5%)
Profit trades: 827 (0.37%)       Loss trades: 1416 (0.63%)
Consecutive wins: 6             Consecutive losses: 13
Average profit trade: 1086.35    Average loss trade: -544.57
Max win: 5566.4                 Max loss: -4642.83
```

Evolución del capital en el periodo optimizado



*El reporte completo se puede encontrar en la carpeta “reportes” del archivo de entrega.

Validación

Una vez obtenidos los resultados del genético los nuevos parámetros de la estrategia son:

- _ema_fast=22
- _ema_slow=35
- _williams_period=4
- _williams_upperband=-10
- _williams_lowerband=-91
- _stoploss=169
- _takeprofit=363

Tras el proceso de optimización se pasará a validar la estrategia con los nuevos parámetros. Una vez realizado el backtest en el periodo de 01/01/2019 hasta el 30/08/2019 los resultados son los siguientes:

```
                    *** BACKTEST REPORT ***
_____________________________________________________________________

------------------- Strategy parameters --------------------
EMA Fast: 22
EMA Slow: 35
Williams %R period: 4
Williams %R upperband: -10
Williams %R lowerband: -91
Stop loss: 169
Take profit: 363
Lot size: 3

--------------------- General Results ----------------------

Initial portfolio value: 100000.00
Final portfolio value: 114880.66
Total gross profit: 17937.00
Total netprofit: 14873.96
Commisions: 3063.04
Total return: 14.88%
Expected payoff: 734.25

------------------------- Drawdown --------------------------

Drawdown: 2.63%              Absolute Drawdown: 3107.54
Max drawdown: 6.47%          Max absolute drawdown: 6692.34

-------------------------- Trades ---------------------------

Total trades: 182
Long trades: 86 (0.47%)          Short trades: 95 (0.52%)
Profit trades: 69 (0.38%)        Loss trades: 112 (0.62%)
Consecutive wins: 6              Consecutive losses: 7
Average profit trade: 1076.15    Average loss trade: -530.18
Max win: 1405.32                 Max loss: -964.7
```

Como muestran los resultados de la validación, los resultados de esta combinación de parámetros obtenidos por el algoritmo genético son suficientemente sólidos por lo que aquí concluye con éxito la primera optimización de la estrategia inicial.

Overfitting

No obstante, cabe mencionar que el proceso de optimización se ha realizado varias veces para comprobar su funcionamiento dando resultados de todo tipo, desde modelos completamente sobreajustados al periodo de optimización que no son capaces de predecir nada en la validación, hasta modelos que continúan de forma robusta con sus resultados.

Por esto, el proceso de validación que se ha realizado resulta imprescindible. Aun así, un mejor proceso de validación a lo largo de la optimización seria lo indicado para controlar que el algoritmo genético no puntué positivamente soluciones que están sobre ajustadas. Este proceso no formara parte de este trabajo, pero es un aspecto que merecerá la continuación de su estudio en futuras investigaciones.

Optimización 2: Red neuronal

La segunda fase de la optimización de la estrategia consistirá en la creación de una red neuronal capaz de discriminar las buenas operaciones de las malas operaciones, basándose en distintas características.

La elección de este método de clasificación se debe a que es uno de los modelos más potentes y con mayor capacidad de aprendizaje actualmente.

Datos

Para el entrenamiento de la red neuronal utilizaremos el rango de datos desde 2009 hasta 2019 (ambos incluidos) de los 4 pares de divisas.

Se hará una división del set de datos de la siguiente manera:

- Datos de entrenamientos: 70%
- Datos de validación: 20%
- Datos de test: 10%

Para preparar los datos de cara a poder ser utilizados en el entrenamiento del modelo se ha elaborado el notebook “TFG_data_preparation” que se encuentra en la carpeta “código” del archivo de entrega. En este notebook se preparan los datos de cada par de divisas de forma individual. Se han utilizado los parámetros definidos por el algoritmo genético para el cálculo de los indicadores técnicos de la estrategia, así como los valores de stoploss y takeprofit. Los pasos que se dan en esta primera preparación son los siguientes:

- Establecer la fecha en formato datetime como índice.
- Se añaden los indicadores que utiliza la estrategia para ser utilizados como inputs del modelo.
- Se añaden las distintas señales que calcula el algoritmo para las entradas, para ser utilizadas como inputs del modelo.
- Se añaden las señales de compra y venta para ser utilizadas como inputs del modelo.
- Para que el modelo tenga más características se han añadido 20 medias móviles exponenciales como características que el modelo pueda tomar como inputs.

- Se reduce el set de datos a las entradas que tengan señal de compra o venta. Son las únicas entradas que necesitamos para entrenar el modelo ya que queremos que aprenda a discriminar que entrada será buena y cual será mala, en el momento en que se de una señal de compra o de venta.
- Por último, se añaden etiquetas a todas las entradas. Para ello se ha calculado para cada orden cual ha sido el primer precio alcanzado: El precio de "stoploss" o el precio de "takeprofit". Las etiquetas contendrán los valores binarios 0 y 1, siendo 0 equivalente a una mala operación y 1 a una buena operación.
- Se guarda el set de datos etiquetados en formato ".csv" en el directorio de trabajo.

Preprocesado para Keras

El modelo será construido con la librería de Deep learning Keras bajo el backend de Tensorflow.

De cara a entrenar el modelo de Keras con el set de datos anteriormente etiquetado se tendrá que preprocesar la información para una mayor facilidad en el proceso de entrenamiento. Los datos se preparan para un problema de clasificación. El preprocesado se encuentra en el notebook "TFG_neural_network" en la carpeta "código" del archivo de entrega.

En esta última preparación del set de datos se realizan los siguientes ajustes:

- Se unifican en un solo set de datos los activos con los que se quiere entrenar el modelo, en nuestro caso: EURUSD, GBPUSD, AUDUSD, NZDUSD.
- Se reordenan todas las entradas del nuevo set de datos conjunto.
- Se separa el set de datos en características y etiquetas.
- Se normalizan todas las características con valores comprendidos entre 0 y 1.
- Se separan las características y las etiquetas en un set de entrenamiento y otro set de test para finalmente evaluar la robustez del modelo. El set de datos de test estará formado por el 10% del set de datos general.

- Se convierten las etiquetas del set de datos de entrenamiento a categóricas.

De esta forma el set de datos quedará listo para entrenar el modelo que será creado en Keras.

Red neuronal en Keras

La arquitectura del modelo se basará en una red neuronal densa multicapa que tratará de resolver el problema de clasificación y se entrenará mediante aprendizaje supervisado.

El código del modelo de Keras se encuentra en el notebook "TFG_neural_network" en la carpeta "código" del archivo de entrega.

Como input recibirá las características anteriormente preparadas. La función de activación elegida para este problema será la función "relu" debido a que es la que mejor funciona con el algoritmo "descenso del gradiente" usado para calcular el error durante el entrenamiento en este tipo de redes. Sin embargo, la función de activación de la capa de salida será "softmax" ya que estamos tratando de resolver un problema de clasificación y es la que mejor desempeño tiene en la última capa en este tipo de problemas.

Para compilar el modelo se han utilizado como hiperparámetros iniciales:

- Un learning rate de 0,00001.
- Adam como optimizador.
- La función de coste "categorical_crossentropy".
- Para las métricas se utilizará el "accuracy".

Para validar el modelo se ha usado un Split del 20%.

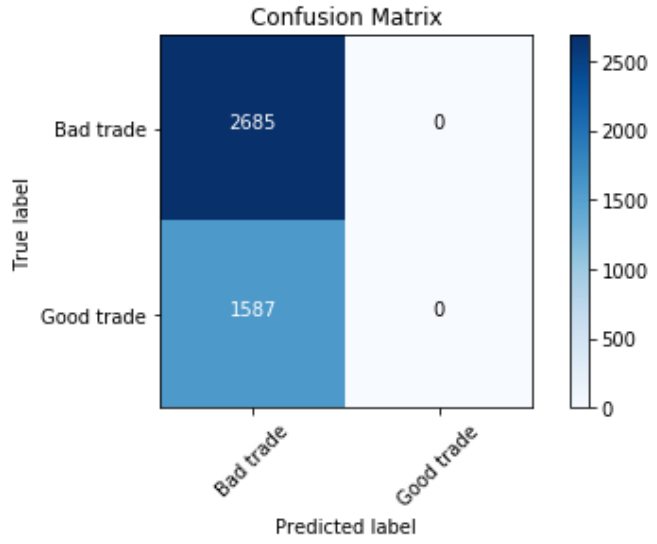
La arquitectura del modelo queda resumida de esta manera:

Layer (type)	Output Shape	Param #
dense_33 (Dense)	(None, 50)	1650
dense_34 (Dense)	(None, 100)	5100
dense_35 (Dense)	(None, 100)	10100
dense_36 (Dense)	(None, 100)	10100
dense_37 (Dense)	(None, 2)	202

Total params: 27,152
Trainable params: 27,152
Non-trainable params: 0

Los resultados de entrenar el modelo durante 50 épocas del modelo creado y entrenado se resumen en la siguiente matriz de confusión:

```
Confusion matrix, without normalization
[[2685    0]
 [1587    0]]
```



Como se puede observar el modelo clasifica como malas operaciones todas las operaciones con lo que se asegura un resultado positivo pero que no nos es de ninguna utilidad ya que si el algoritmo tacha todas las potenciales operaciones como malas nunca realizara ninguna operación.

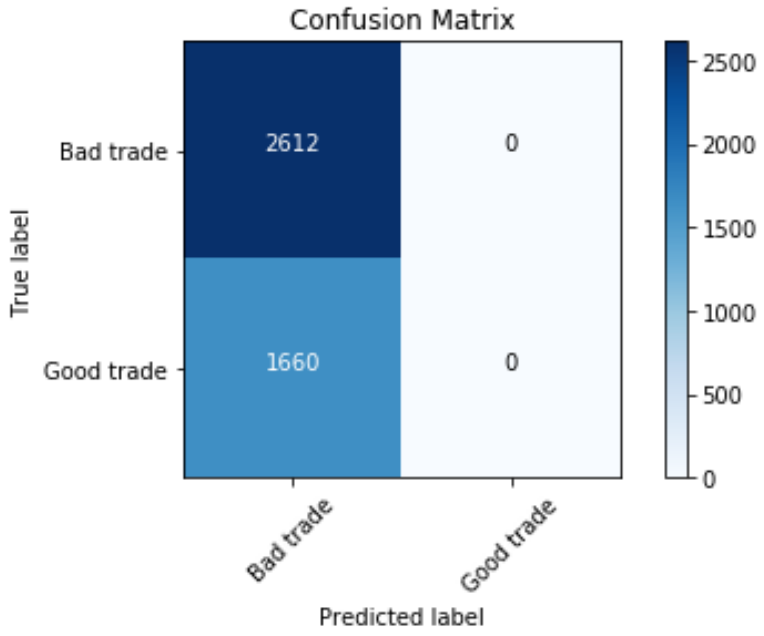
Se procederá a crear una arquitectura nueva para nuestra red neuronal aumentando tanto capas como nivel de neuronas.

La nueva arquitectura queda resumida de la siguiente manera:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_23 (Dense)             (None, 200)               6600
_________________________________________________________________
dense_24 (Dense)             (None, 500)               100500
_________________________________________________________________
dense_25 (Dense)             (None, 500)               250500
_________________________________________________________________
dense_26 (Dense)             (None, 500)               250500
_________________________________________________________________
dense_27 (Dense)             (None, 500)               250500
_________________________________________________________________
dense_28 (Dense)             (None, 500)               250500
_________________________________________________________________
dense_29 (Dense)             (None, 500)               250500
_________________________________________________________________
dense_30 (Dense)             (None, 500)               250500
_________________________________________________________________
dense_31 (Dense)             (None, 500)               250500
_________________________________________________________________
dense_32 (Dense)             (None, 2)                 1002
=================================================================
Total params: 1,861,602
Trainable params: 1,861,602
Non-trainable params: 0
_________________________________________________________________
```

Los resultados de entrenar esta red neuronal durante 50 épocas se han evaluado con la partición del set de datos test y quedan resumidos en la siguiente matriz de confusión:

```
Confusion matrix, without normalization
[[2612    0]
 [1660    0]]
```



Como se puede observar, a pesar de haber aumentado la capacidad del modelo este proporciona resultados prácticamente idénticos y sigue clasificando todas las potenciales operaciones como malas.

El problema puede deberse a una errónea elección de características a la hora de entrenar el modelo o a un problema demasiado complejo como para que pueda ser explicado con los datos de los que se dispone.

Debido al nulo avance de esta idea de usar una red neuronal para discriminar entradas a mercado se toma la decisión de no incluirla finalmente en el código de la estrategia ya que el algoritmo jamás realizaría operaciones.

Conclusión

En este trabajo se ha creado una estrategia desde cero para negociar en los mercados financieros y se ha tratado de optimizarla mediante dos técnicas diferentes de inteligencia artificial: Un algoritmo genético perteneciente a la categoría de algoritmos evolutivos y una red neuronal densa de aprendizaje supervisado perteneciente al campo del Deep learning.

La fase de optimización mediante el uso del algoritmo genético ha dado resultados satisfactorios y ha conseguido generar un modelo robusto. No obstante, también ha mostrado debilidades que puede tener la optimización a través de estos algoritmos evolutivos con los que hay que ser precavido para no sobre ajustar el modelo en la optimización.

En la fase de optimización mediante una red neuronal se ha intentado crear un modelo de machine learning capaz de discriminar entradas a mercado mediante aprendizaje supervisado. Desafortunadamente los resultados obtenidos tras crear y evaluar esta idea no han sido los esperados. El modelo creado clasifica todas las operaciones como “malas” por lo que finalmente se ha decidido no incluir el modelo en la estrategia con el objetivo de no entorpecer la optimización conseguida con el algoritmo genético.

Líneas de trabajo futuras

Tras analizar los resultados obtenidos, se pueden establecer como líneas futuras de la investigación las siguientes:

- **Backtest**:
  - Mejorar la eficiencia de la función de backtest con código propio para minimizar su duración. Esto permitiría evaluar un mayor número de soluciones y en menor tiempo con el algoritmo genético.
  - Mejorar los reportes gráficos para una representación más limpia.
- **Algoritmo genético**:
  - Implementar un método de validación durante la propia optimización. Esto ayudaría a avanzar hacia soluciones mas robustas. Una posible opción seria evaluar no solo la solución generada, sino las de su alrededor, con el objetivo de descartar aquellas que arrojen un resultado positivo por mera casualidad.
  - Cambiar el ratio de evaluación. Actualmente se utiliza como medidad de desempeño la rentabilidad alcanzada, pero esta medida deja fuera elementos tan importantes como el riesgo. Seria interesante probar diferentes ratios que incluyan el riesgo u otros elementos como el ratio de Sharpe.
- **Red neuronal**:
  - Balancear el set de entrenamiento. Esto impediría que la red opte por elegir siempre la categoría que mayor numero de entradas tenga en el data set de entrenamiento.
  - Estudiar la calidad explicativa de los inputs elegidos para entrenar el modelo. Es probable que muchos sean repetitivos y falten nuevos inputs de mayor calidad explicativa.
  - Reducir drásticamente el modelo de predicción. Los 2 modelos de red planteados son enormes y contienen demasiados parametros a entrenar en comparación con el reducido numero de datos que se les proporciona. Reducir el modelo o cambiarlo a un modelo más sencillo podría aumentar las probabilidades de un aprendizaje exitoso.

- Generar mas datos de forma artificial. Esta técnica podría ayudar a entrenar modelos grandes. Los datos financieros son limitados y llegado un punto no se puede sacar de donde no hay.

Configuración del entorno

El código está diseñado para mostrarse y ser utilizado mediante Jupyter Notebook.

Librerías

- backtrader==1.9.78.123
- matplotlib==3.8.1
- reportlab==4.0.6
- pandas==2.1.2
- numpy==1.26.1
- ta==0.11.0
- Keras==2.14.0
- tensorflow==2.14.1
- scikit-learn==1.3.2