



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Estrategias de inversión en bolsa basadas
en la detección de patrones mediante
análisis técnico

*Stock investment strategies based in the detection of patterns
through technical analysis*

Aitor Alonso Melián

La Laguna, 13 de julio de 2023

D. **Cándido Caballero Gil**, con N.I.F. 42.201.070-A profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Javier Giner Rubio**, con N.I.F. 21.649.507-K profesor Titular de Universidad adscrito al Departamento de Economía, Contabilidad y Finanzas de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

"Estrategias de inversión en bolsa basadas en la detección de patrones mediante análisis técnico"

ha sido realizada bajo su dirección por D. **Aitor Alonso Melián**, con N.I.F. 51.148.286-N.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 13 de julio de 2023

Agradecimientos

Quiero aprovechar este espacio para expresar mi agradecimiento a mi tutor, Cándido Caballero Gil, por su orientación y compromiso con el trabajo, y también a mi cotutor, Javier Giner Rubio, por su aportación y consejo al proyecto. También me gustaría agradecer a mi familia y amigos, por todo el apoyo que me han dado durante, no solo el desarrollo de este proyecto, si no de la carrera en general. Probablemente sin ellos no habría conseguido llegar tan lejos.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-
NoComercial-CompartirIgual 4.0 Internacional.

Resumen

Este trabajo de fin de grado se centra en el desarrollo de un programa de detección de patrones chartistas en bolsa basado en análisis técnico. Todo esto se hace con el objetivo de brindar a los inversores una herramienta eficiente y dar precisión para la toma de decisiones en el mercado financiero. El análisis técnico se fundamenta en el estudio del comportamiento pasado de los precios de las acciones, buscando patrones repetitivos que puedan ofrecer indicios sobre futuros movimientos del mercado. Esta herramienta hace uso del algoritmo Dynamic Time Warping para comparar series temporales entre sí, y tratar de detectar patrones. Tras encontrar patrones con una distancia suficientemente cercana a los patrones predefinidos, estos son revisados para comprobar si efectivamente se parecen al patrón que se intenta identificar. Una vez identificados los patrones, se comprobarán cuantos patrones se cumplen, frente a cuantos no se cumplen, dando como resultado el porcentaje de cumplimiento de cada patrón.

Palabras clave: Análisis técnico, inversión, bolsa, patrones chartistas.

Abstract

This bachelor's thesis focuses on the development of a program for detecting chartist patterns in the stock market based on technical analysis. All this is done with the aim of providing investors with an efficient tool and precision for decision-making in the financial market. This final degree project focuses on the development of a program for detecting chartist patterns in the stock market based on technical analysis. All this is done with the aim of providing investors with an efficient tool and precision for decision-making in the financial market. The technical analysis is based on the study of the past behavior of the prices of the actions, looking for repetitive patterns that can offer indications about future movements of the market. This tool makes use of the Dynamic Time Warping algorithm to compare time series with each other, and try to detect patterns. After finding patterns with a sufficiently close distance to the predefined patterns, these are reviewed to see if they actually resemble the pattern that is trying to be identified. Once the patterns have been identified, it will be checked how many patterns are met, versus how many are not met, resulting in the percentage of compliance of each pattern.

Keywords: Technical analysis, investment, stock market.

Índice general

1. Introducción	1
1.1. Motivo del proyecto	1
1.2. Objetivos	1
1.3. Fases del proyecto	2
2. Contexto	3
2.1. Conceptos relevantes	3
2.1.1. Análisis técnico	3
2.1.2. Análisis chartista	3
2.1.3. Backtesting	5
2.1.4. Dynamic Time Warping	6
2.2. Antecedentes y estado actual del tema	6
3. Desarrollo de proyecto	8
3.1. Estado previo de PerseumAI	8
3.2. Implementación de mejoras	10
3.2.1. Mejoras en la división de ventanas	10
3.2.2. Líneas para detectar la forma de la figura en el gráfico	11
3.2.3. Filtrado por distancias para reducir la carga	11
3.2.4. Implementación de la media móvil simple	12
3.2.5. Mejoras en el uso del Dynamic Time Warping	12
3.2.6. Mejoras en la detección de patrones individuales	13
3.2.7. Mejoras en la interfaz	14
3.2.8. Cambios en la detección de patrones actuales.	15
3.3. Nuevos patrones implementados	16
3.3.1. Hombro cabeza hombro	17
3.3.2. Hombro cabeza hombro invertido	19
3.3.3. Triángulo ascendente	20
3.3.4. Triángulo descendente	23
3.4. Distribución de la aplicación	24
3.5. Algoritmos adicionales	24
4. Resultados obtenidos	26
4.1. Desempeño de la herramienta en datos históricos	26
4.1.1. Sector tecnológico	26
4.1.2. Sector financiero	27
4.1.3. Análisis de resultados	28
4.2. Selección de parámetros	28
4.2.1. Tamaño de ventana	28

4.2.2. Longitud de media móvil	28
4.2.3. Cantidad de datos de ejemplo seleccionados	29
4.2.4. Distancia para filtrar	29
4.3. Análisis de errores, limitaciones y posibles mejoras	30
4.3.1. Doble techo VS hombro cabeza hombro	30
4.3.2. Triángulos similares a canales o cuñas	31
5. Conclusiones y líneas futuras	33
5.1. Conclusiones	33
5.2. Líneas de trabajo futuras	34
6. Summary and Conclusions	35
6.1. Conclusions	35
7. Presupuesto	36
7.1. Costes materiales	36
7.2. Costes de trabajo	36
A. Elementos adicionales	37
A.1. Enlaces al programa	37
A.2. Fuentes de datos de las tablas	37
A.2.1. Resultados hombro cabeza hombro sector tecnológico	40
A.2.2. Resultados hombro cabeza hombro sector financiero	50

Índice de Figuras

2.1.	Hombro-cabeza-hombro	4
2.2.	Hombro-cabeza-hombro invertido	4
2.3.	Triangulo descendente	5
2.4.	Triangulo ascendente	5
2.5.	Distancia Euclidea vs DTW	6
3.1.	Interfaz	9
3.2.	Doble techo	11
3.3.	Media móvil vs Precio Actual	12
3.4.	Antes vs Después	13
3.5.	Doble techo	13
3.6.	Nueva interfaz	15
3.7.	Hombro cabeza hombro componentes	17
3.8.	HCH con el objetivo cumplido	18
3.9.	HCH con el objetivo no cumplido	18
3.10.	HCH invertido con el objetivo cumplido	19
3.11.	HCH invertido con el objetivo no cumplido	20
3.12.	Triangulo ascendente componentes	21
3.13.	Triangulo ascendente con objetivo cumplido	22
3.14.	Triangulo ascendente con objetivo no cumplido	22
3.15.	Triangulo descendente con objetivo cumplido	23
3.16.	Triangulo descendente con objetivo no cumplido	24
4.1.	Doble techo más similar a hombro cabeza hombro	31
4.2.	Triangulo descendente ambiguo	32
A.1.	Datos financieros	37
A.2.	Datos financieros	38
A.3.	Datos tecnológicos	39
A.4.	Datos tecnológicos	40

Índice de Tablas

4.1. Conjunto de empresas del sector tecnológico	26
4.2. Resultados de 'PerseumAI' de las empresas del sector tecnológico (A.3) . .	27
4.3. Conjunto de empresas del sector financiero	27
4.4. Resultados de 'PerseumAI' de las empresas del sector financiero (A.1) . . .	28
4.5. Distancias medias según DTW de los ejemplos de los patrones entre sí . . .	30
7.1. Tabla de costes materiales	36
7.2. Tabla de costes de trabajo	36

Capítulo 1

Introducción

1.1. Motivo del proyecto

Este trabajo tiene como objetivo principal continuar el desarrollo de una herramienta programada en Python para la detección de patrones en bolsa mediante análisis técnico. El análisis técnico es el uso de datos históricos del mercado con el fin de identificar patrones y tendencias que puedan ayudar a tomar decisiones de inversión más informadas.

El mercado de valores es un entorno altamente complejo en el cual los inversores buscan constantemente identificar oportunidades para maximizar beneficios y gestionar riesgos. La detección de patrones en bolsa es una herramienta utilizada en el análisis técnico para tratar de identificar señales de compra y venta potenciales basadas en la repetición histórica de comportamientos del mercado.

Gracias a estar desarrollado en Python, el programa hace uso de una amplia gama de bibliotecas y herramientas especializadas en el análisis de datos, como lo es 'Pandas'. El desarrollo de programa no solo implica el diseño e implementación de algoritmos de detección de patrones, si no también la recopilación y el procesamiento de datos históricos del mercado, así como la visualización de resultados.

Este proyecto permitirá a los usuarios analizar datos históricos de los precios de las acciones, identificar patrones relevantes, con el objetivo de servir como apoyo para la toma de decisiones de inversión.

1.2. Objetivos

El objetivo de este proyecto es continuar con el desarrollo de la herramienta creada por el compañero (Gabriel García, 2022), que realizó la primera parte de este proyecto el curso anterior. En este trabajo se pretende ampliar el comportamiento de la herramienta añadiendo más patrones a identificar, a la vez que implementar, potenciales mejoras al programa.

Algunas de estas mejoras son:

- Implementar los patrones chartistas hombro cabeza hombro, hombro cabeza hombro invertido, triángulo ascendente y triángulo descendente.
- Mejorar la detección de patrones suavizando los datos utilizando la media móvil simple.

- Cambiar el uso de las ventanas en el programa, de manera que no pueda quedar ningún patrón intermedio entre ventanas.
- Hacer el programa más versátil dándole la posibilidad al usuario de cambiar ciertos parámetros, como por ejemplo el tamaño de la ventana, la fecha de finalización, etcétera.
- Mejorar la interfaz para que sea más usable y accesible.
- Ofrecer un porcentaje de cuántos patrones han cumplido su objetivo.

1.3. Fases del proyecto

En esta sección se especifican las diferentes fases de desarrollo del proyecto:

1. Investigación acerca del algoritmo Dynamic Time Warping, el cual es utilizado por la herramienta desarrollada para detección de patrones.
2. Investigación sobre la herramienta FinViz, de la cual se puede extraer datos de patrones que se están formando en la actualidad.
3. Familiarización con el código de PerseumAI.
4. Implementación de posibles mejoras en la detección de patrones existente.
5. Investigación sobre los diferentes tipos de patrones a implementar.
6. Recogida de datos de los patrones a implementar.
7. Implementación del código para la detección de los patrones seleccionados.
8. Una vez finalizadas las mejoras y la implementación de patrones nuevos en la herramienta, realizar mejoras gráficas en la interfaz.
9. Por último, desarrollar la memoria que describa el desarrollo del proyecto y la investigación realizada.

Capítulo 2

Contexto

2.1. Conceptos relevantes

En este capítulo se explicarán algunos de los conceptos relevantes para el proyecto, como por ejemplo, que es el análisis técnico, cuales son sus diferencias con el análisis chartista, y que es el backtesting. También se explicarán los patrones que reconoce la herramienta. El capítulo finalizará con una revisión de los antecedentes del tema tratado en el trabajo.

2.1.1. Análisis técnico

'El análisis técnico es el proceso de analizar los precios históricos de un valor en un esfuerzo encaminado a determinar los precios futuros probables. Esto se hace comparando la actuación actual de los precios (y, por lo tanto, las expectativas actuales) con los movimientos semejantes en el pasado, para prever una salida lógica a la situación actual.' (Achelis, 1995).

Los analistas técnicos creen que todos los factores fundamentales que afectan a un activo ya están reflejados en su precio, por lo que se centran exclusivamente en el estudio de los gráficos y patrones de precios. Utilizan herramientas como indicadores técnicos (por ejemplo, medias móviles, osciladores, bandas de Bollinger) y líneas de tendencia para identificar patrones repetitivos y señales de compra o venta. El análisis técnico busca aprovechar las tendencias y los puntos de inflexión en los precios.

2.1.2. Análisis chartista

El análisis chartista es una forma específica de análisis técnico que se centra en el estudio de los gráficos de precios para identificar patrones y tendencias. Los analistas chartistas creen que los precios se mueven en patrones que pueden ayudar a predecir futuros movimientos de precios. Algunos ejemplos de patrones chartistas incluyen triángulos ascendente o descendente, hombro-cabeza-hombro, banderas, etc. Los analistas chartistas también prestan atención a los niveles de soporte y resistencia, que son niveles de precios en los que hay altas probabilidades que el precio se detenga o se revierta. En los siguientes apartados se expondrán algunas de las figuras más extendidas en el análisis chartista.

'El **hombro cabeza hombro** es una figura compuesta por tres techos o máximos consecutivos, siendo la cabeza la mayor. La línea recta que se traza entre los dos mínimos de la cabeza se le denomina 'cuello'. Este patrón indica de forma bastante fiable el cambio

de la tendencia alcista a bajista, provocada por un agotamiento gradual de la tendencia primaria, vinculado a un proceso de distribución.’(Torre Gallegos, A.d.l. y Ruíz Martínez, R.J. (1999), p. 17)

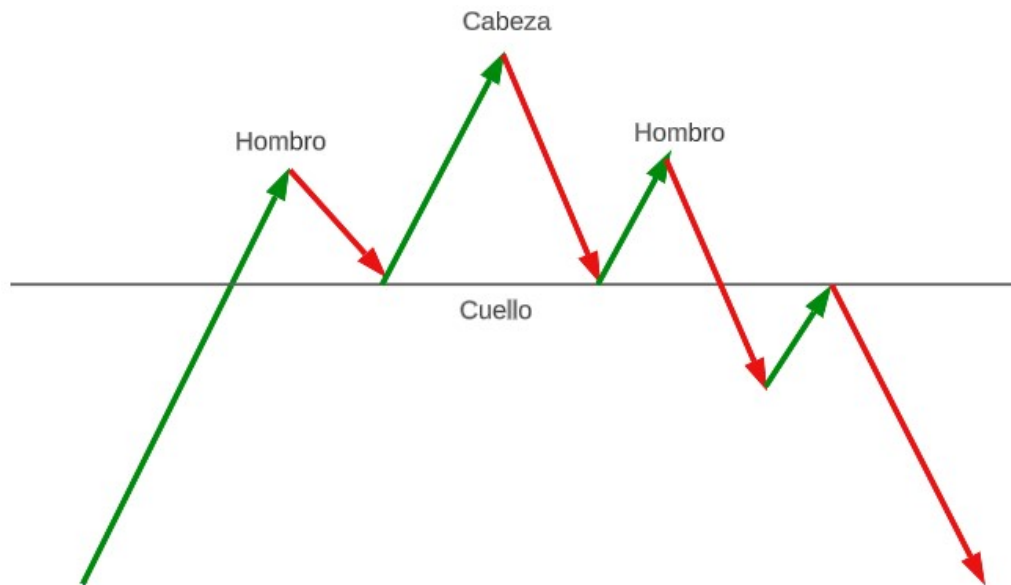


Figura 2.1: Hombro-cabeza-hombro

También existe el hombro cabeza hombro invertido, el cual indica lo contrario, es decir, el cambio de una tendencia bajista a alcista.

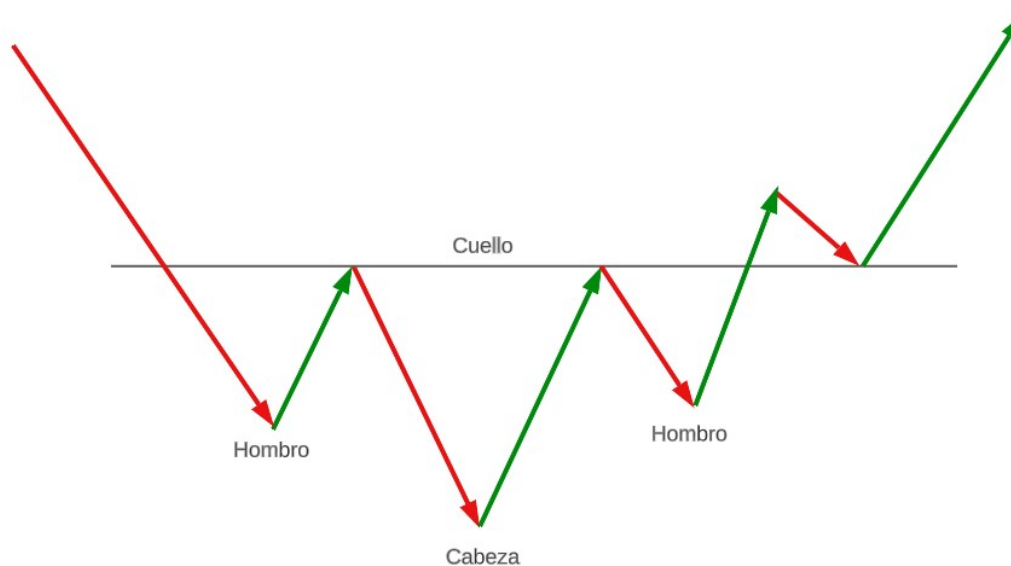


Figura 2.2: Hombro-cabeza-hombro invertido

El objetivo de los patrones en el análisis chartista es ser capaces de identificar un cambio de tendencia, o la continuación de la misma. Hasta ahora se han mostrado patrones de cambio de tendencia, pero también existen patrones de continuación de tendencia, como por ejemplo, el triángulo ascendente y descendente.

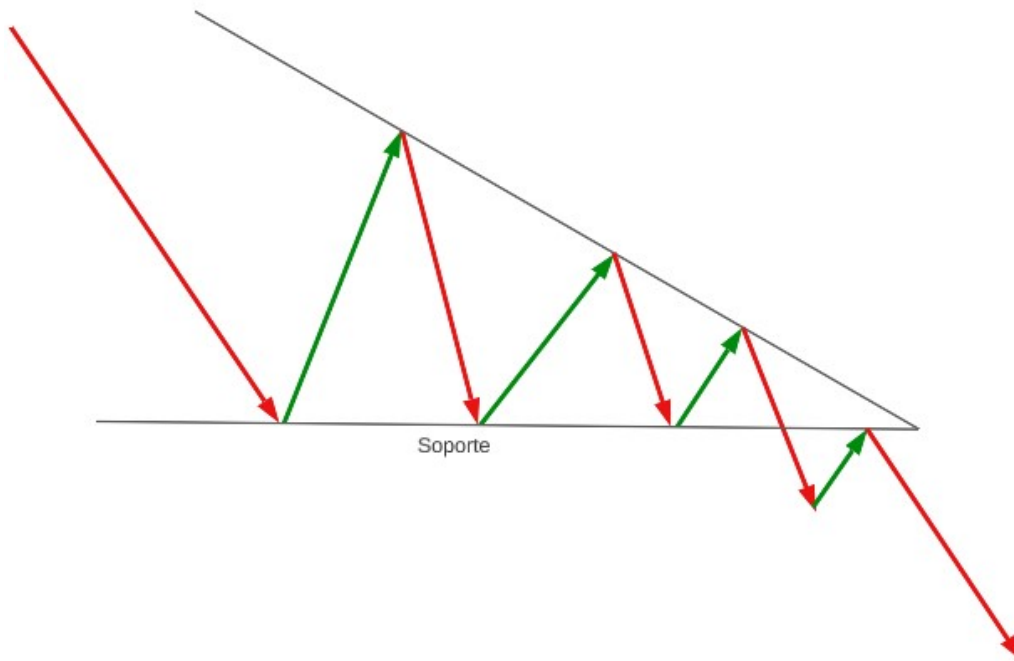


Figura 2.3: Triangulo descendente

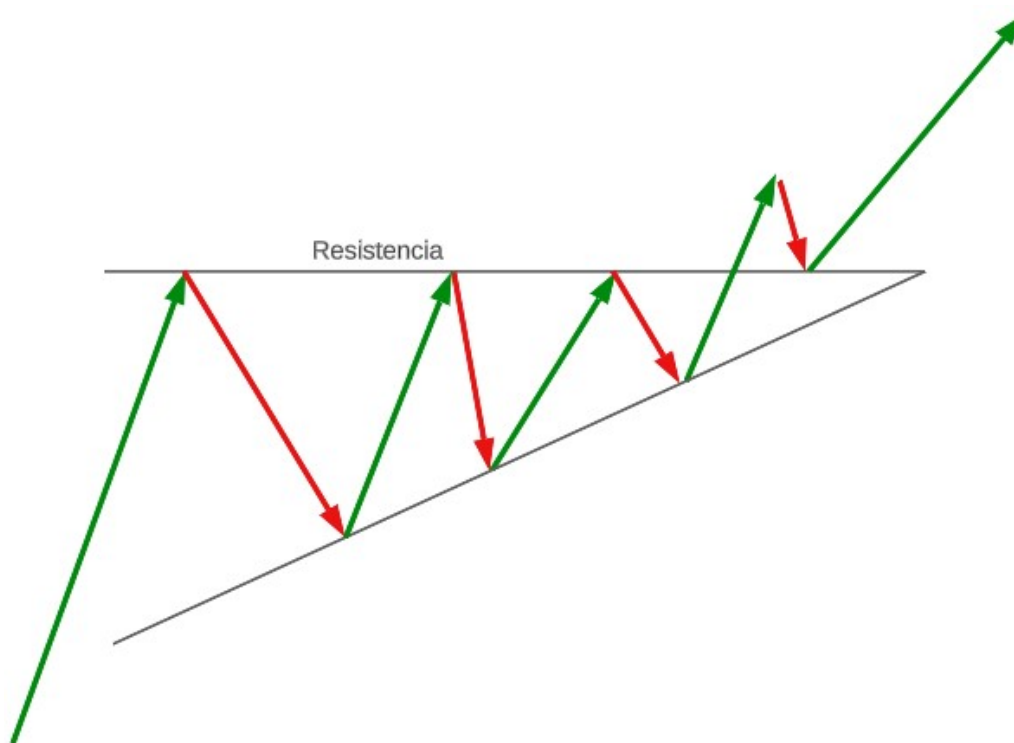


Figura 2.4: Triangulo ascendente

2.1.3. Backtesting

'El backtesting, también conocido como prueba de sistemas, es el concepto de tomar una estrategia y retroceder en el tiempo para comprobar que habría sucedido si la estrategia se hubiese seguido fielmente. Asumiendo que si la estrategia ha funcionado previamente, tiene una buena, pero no segura probabilidad de funcionar nuevamente en el futuro, y por el contrario, si el concepto no ha funcionado bien en el pasado, probablemente no funcionará bien en el futuro.' (Jirai y Chengqi, 2005)

La idea es que gracias a la herramienta desarrollada, se puede comprobar la estrategia de invertir en base a identificación de patrones, como los mencionados en el apartado anterior. Se pueden evaluar la eficacia de los diferentes patrones en diferentes momentos temporales, y para diferentes conjuntos de empresas. Por ejemplo, se podría comprobar que porcentaje de hombros cabeza hombro se cumplen para empresas del sector tecnológico en el periodo 2008-2015, o cómo se comportaron las empresas del Rusell 2000 entre 1999 y 2005 por ejemplo.

2.1.4. Dynamic Time Warping

El algoritmo Dynamic Time Warping es un algoritmo cuyo objetivo es medir la similitud entre dos secuencias de datos temporales. La diferencia con otros algoritmos de este mismo propósito (Como la distancia euclídea) es que se pueden obtener similitudes incluso frente a un desfase en la velocidad o en el tiempo de las series.

Este algoritmo es considerablemente mejor que la distancia euclídea para el propósito del trabajo, ya que los patrones no siempre se van a formar de la misma manera, ya que pueden haber diferencias en la amplitud, longitud, o incluso en el tiempo de los patrones.

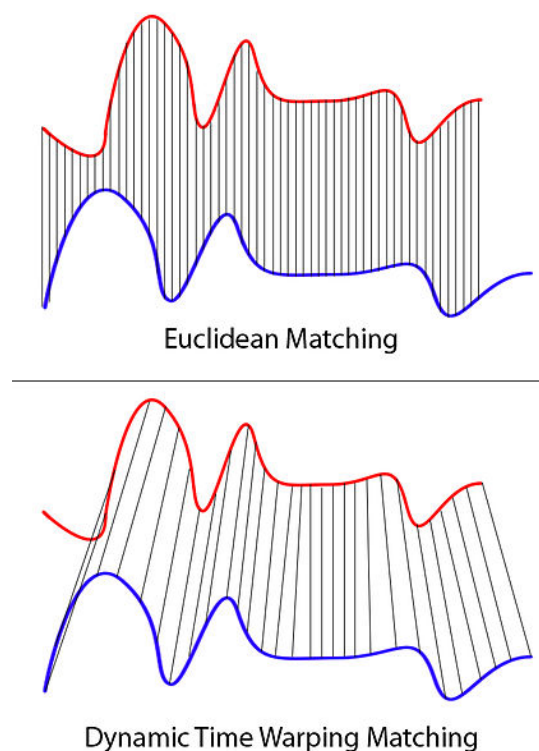


Figura 2.5: Distancia Euclídea vs DTW

2.2. Antecedentes y estado actual del tema

Según (Farias et al, 2017), se han hecho una gran cantidad de estudios en los últimos años sobre análisis técnico, pero muy pocos de ellos se han dedicado a tratar de consolidar

todo el conocimiento disponible sobre este. Por ello, este meta análisis trata de recopilar los estudios más relevantes del ámbito del análisis técnico, para así tratar de sacar una conclusión de todos ellos. En el ámbito del análisis chartista, que es en el que se centra este trabajo, este meta análisis encuentra especialmente poco común artículos dedicados a este tema. Concretamente menciona que, de los 85 artículos seleccionados, sólo 2 de ellos hacen referencia al análisis chartista. Por esta razón, este trabajo puede ser interesante, ya que explora una subcategoría no tan conocida en el análisis técnico.

Por otro lado, existe un artículo que describe una herramienta bastante similar a la que se está desarrollando en este trabajo. (Sang Hyuk et al, 2018) describen un sistema basado en identificación de patrones realizado con el algoritmo 'Dynamic Time Warping', que trata de reconocer patrones en los datos del mercado por la mañana, para luego construir una estrategia para la tarde. Según su artículo, consiguen unos resultados estables y efectivos con relativamente poca frecuencia en las operaciones de trading.

También existen otros sistemas como (Fu-Lai et al, 2004) que hacen referencia a la dificultad de procesar, analizar y extraer información de los datos de series temporales debido a su naturaleza numérica y continua. Argumentan que, segmentar los datos con una longitud fija es un enfoque demasiado simplificado para este problema, por lo que proponen utilizar inteligencia artificial, concretamente la rama de computación evolutiva, para tratar de determinar la manera correcta de segmentar los datos. Según describen, han obtenido buenos resultados utilizando este algoritmo en datos obtenidos de la bolsa de Hong Kong.

Por otra parte, (Dong, Zhou, 2002) indican haber creado un sistema el cual es capaz de identificar patrones visuales claramente definidos, pero que también es capaz de detectar diferencias sutiles las cuales podrían ser objeto de controversia en el análisis técnico.

El algoritmo Dynamic Time Warping (DTW) no se limita al uso en el ámbito financiero, como el análisis de datos bursátiles. En un estudio, se demostró que el DTW también puede aplicarse al reconocimiento de escritura y gestos (Yu-Liang et al, 2015).

El DTW es capaz de comparar y alinear secuencias temporales de datos, como trazos de escritura o movimientos gestuales, que pueden variar en velocidad y duración. Además, también es utilizado en (Moisés et al, 2022) como una medida de similitud para comparar y alinear las trayectorias de la escritura a mano.

Capítulo 3

Desarrollo de proyecto

3.1. Estado previo de PerseumAI

PerseumAI es una herramienta creada originalmente por Gabriel García, sobre la cual he empezado a trabajar para continuar con su desarrollo. Gabriel realizó una herramienta que detecta, mediante el algoritmo Dynamic Time Warping, el patrón doble techo y doble suelo, en un periodo de tiempo especificado por el usuario.

Todo el funcionamiento que se explique en este apartado, es referente a la herramienta en su estado inicial, sin ninguna modificación. Para funcionar, el programa necesita una serie de parámetros introducidos por el usuario. Estos parámetros son:

1. Patrón a buscar: Hay unos checkbox que permiten seleccionar los patrones que se desea analizar. Se puede seleccionar un solo patrón, o ambos.
2. Año inicial: Se tiene que especificar un año desde el cual partir para buscar los patrones. No se puede especificar un año final, si no que busca hasta la actualidad.
3. Tickers de las empresas a analizar: El botón 'Open File' abre una ventana para seleccionar un fichero de texto que contenga los tickers de las empresas a analizar. El formato debe ser un archivo TXT que tenga los tickers de las empresas separados por saltos de líneas. Se pueden colocar comas o no al final de cada línea, ya que la herramienta detecta los tickers de igual manera.
4. Intensive Search: Un checkbox que si se marca hace que el programa haga una búsqueda mas intensiva sobre los patrones actuales. No afecta a los patrones históricos.

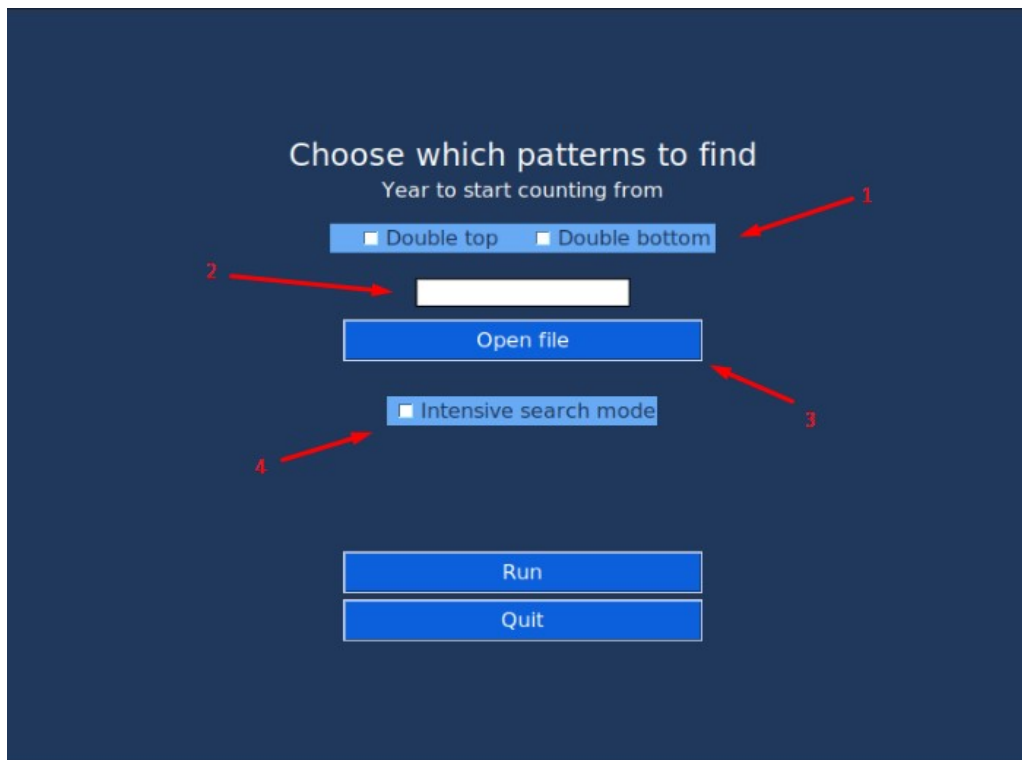


Figura 3.1: Interfaz

El funcionamiento, una vez seleccionados los parámetros es el siguiente:

1. Se escoge la 'ventana' de datos. A partir de ahora, se llamará 'ventana' a un subconjunto de datos de longitud X obtenidos de los datos de cierre totales. Esta longitud X está predefinida en el programa con un valor de 51. Ejemplo: Imaginemos que tenemos todos los precios de cierre de cada día en bolsa de 'Apple' desde 2019 hasta 2023. Eso serían $365 * 4 = 1460$ precios de cierre. Entonces, la herramienta empezará a iterar seleccionando los primeros 51 precios de cierre.
2. Una vez seleccionada la primera ventana, se comienza a iterar para comparar la ventana seleccionada con los patrones guardados mediante Dynamic Time Warping.
3. Se selecciona el patrón de los datos de ejemplo que más se asemeje a la ventana. Esto se logra gracias a que la comparación de Dynamic Time Warping da como resultado una distancia. A menor distancia, mayor similitud existe entre la ventana y el patrón de ejemplo. Es decir, se escoge el patrón con menor distancia.
4. Se divide la ventana en 'subventanas', dividiéndola primero entre 2, luego entre 3, y finalmente entre 4 subventanas. Esto genera un total de 9 subventanas posibles, más la ventana original, que serían 10. Se vuelve a comparar con los datos de ejemplo si existe alguna subventana que tenga una similitud mejor que la ventana original, y en caso de que así sea, se selecciona esa subventana.
5. Una vez seleccionada la mejor candidata, se pasa un filtro para comprobar si cumple con las características de un doble techo/suelo. Se trata de detectar los dos picos máximos o mínimos, y se comprueba si estos picos son similares, en cuyo caso se confirma que se ha encontrado el patrón.

6. Se traza el objetivo de patrón, y se comprueba si después de haber encontrado ese patrón, el objetivo se ha cumplido o no (en un plazo razonable de tiempo).
7. Una vez comprobado si se cumple o no el objetivo, se guarda este patrón para ser mostrado mas adelante, cuando todos los datos hayan sido analizados.
8. Se escoge la siguiente ventana de datos, y se aplican de nuevo los pasos 1-7.
9. Una vez se ha finalizado el análisis de todos los datos, se muestran los resultados a través de la interfaz.

Estos pasos que se mencionan son un pequeño resumen de como funcionaba la herramienta tal y como la había desarrollado Gabriel. Si desea consultar con más detalle su funcionamiento siempre puede consultar su TFG.

Más adelante explicaré con más profundidad las partes que he modificado de la herramienta, y las razones por las que he realizado esas modificaciones.

3.2. Implementación de mejoras

3.2.1. Mejoras en la división de ventanas

Como se ha indicado en la sección titulada 'Estado previo', los datos se subdividen en ventanas de tamaño uniforme, pero esto implica ciertos matices. Por ejemplo, si el tamaño de la ventana es de 60 días, el programa seleccionará inicialmente los primeros 60 días del conjunto de datos (0-59). Si no se detecta ningún patrón dentro de esa ventana de 60 días, la siguiente ventana seleccionada para examinar posibles patrones sería el rango (1-60). ¿Cuáles son las probabilidades de que, al desplazar la ventana tan solo un día hacia adelante, la herramienta encuentre un patrón? Estas probabilidades son bastante bajas. Por lo tanto, una de las primeras mejoras implementadas consistió en aumentar la cantidad de días que la ventana se desplaza, en caso de no encontrar patrones durante la primera búsqueda. Basándome en mi experiencia con la herramienta, concluí que un valor adecuado serían aproximadamente 25 días.

El objetivo principal de este cambio es lograr una mayor eficiencia en la herramienta, ya que analizar cada ventana consume una cantidad considerable de tiempo. Mediante esta modificación, se ha conseguido que la herramienta sea moderadamente más rápida, dado que la mayoría de las veces no se encuentra un patrón significativo.

Además, realicé modificaciones en el desplazamiento de la ventana cuando se encuentra un patrón. Anteriormente, el valor de desplazamiento correspondía al tamaño de la ventana (o subventana) en la que se encontró el patrón. Sin embargo, después de implementar el cambio, dicho valor se ajustará al índice del último valor de la ventana encontrada. Explicaré esto con un ejemplo:

Antes: Al analizar la ventana 0-59, se realiza una búsqueda en las subventanas para determinar si existe un patrón más sólido en alguna de ellas. Supongamos que se encuentra un patrón en la subventana 45-59. La siguiente ventana a analizar sería 15-74.

Después del cambio: Al analizar la ventana 0-59, se realiza una búsqueda en las subventanas para determinar si existe un patrón más sólido en alguna de ellas. Supongamos que se encuentra un patrón en la subventana 45-59. La siguiente ventana a analizar sería 60-119.

Esta modificación resuelve un problema que hacía que, a veces, detectara el mismo patrón varias veces.

3.2.2. Líneas para detectar la forma de la figura en el gráfico

Han sido añadidas a la herramienta líneas que permiten identificar los puntos importantes del gráfico, como por ejemplo, en el doble techo, una línea que une los dos techos detectados. Por otro lado también se ha añadido a todos los patrones la línea objetivo a alcanzar. Estos cambios se pueden ver en la figura 3.2.

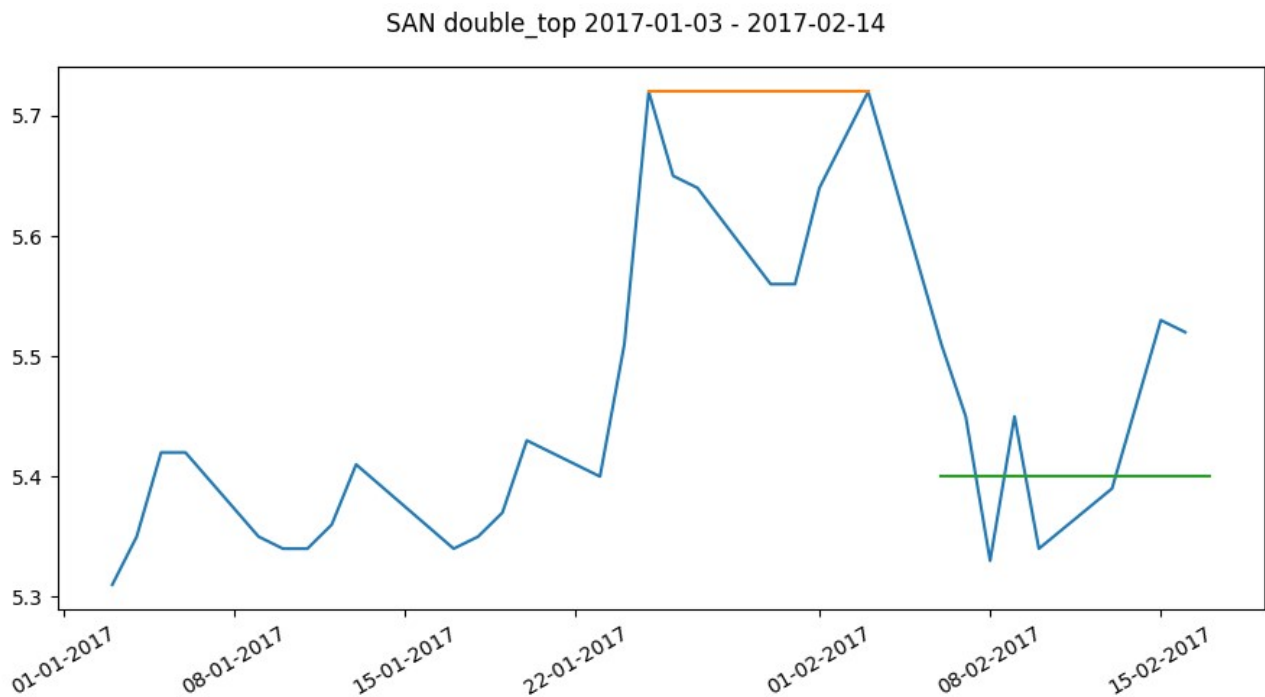


Figura 3.2: Doble techo

Este cambio ha sido añadido para indicar en que parte exacta del gráfico que se ha mostrado se ha encontrado el patrón, ya que puede ser confuso, dado que se presenta una gran parte del gráfico después del patrón encontrado, para revisar si este cumple su objetivo. En este caso (doble techo) es un patrón de cambio de tendencia, y se ve como después de ocurrir, su tendencia cambia.

3.2.3. Filtrado por distancias para reducir la carga

La librería utilizada en el desarrollo de la herramienta 'dtw-python' implementa el algoritmo DTW. Esta librería proporciona, entre otras cosas, la distancia entre dos patrones, es decir, cuan similares son los patrones según el algoritmo Dynamic Time Warping. A menor distancia exista entre dos patrones, más similares serán entre sí. Una de las mejoras que implementé en el código, es descartar patrones cuya distancia fuera considerablemente alta, para así evitar que ese patrón se siga evaluando más allá del algoritmo Dynamic Time Warping. El objetivo de este cambio es, al igual que otros, hacer que el programa tenga menos carga de trabajo, y sea ligeramente más rápido.

3.2.4. Implementación de la media móvil simple

Como se ha comentado en apartados anteriores, el algoritmo DTW se usa para comparar series temporales entre sí, pero en estas series temporales suele haber ruido, el cual puede ser problemático para la comparación de patrones.

Una manera de reducir este ruido es utilizando la media móvil simple. La media móvil es un cálculo utilizado para analizar un conjunto de datos en modo de puntos para crear series de promedios. Así las medias móviles son una lista de números en la cual cada uno es el promedio de un subconjunto de los datos originales. A continuación se mostrará una figura comparando los valores de una media móvil frente a los datos originales.



Figura 3.3: Media móvil vs Precio Actual

La línea azul representa la media móvil de longitud 9, mientras que la línea verde representa el precio de cierre de cada día sin modificar.

La figura no solo muestra las líneas móviles frente al precio de cierre, si no que además se trata de un ejemplo de triángulo ascendente.

Como se puede apreciar, la media móvil reduce el ruido de los datos originales, haciendo que sea más fácil identificar tendencias, además, uno de los estudios mencionados en el apartado de antecedentes ([7] Yu-Liang et al, 2015), hace uso de esta misma técnica con el mismo objetivo.

3.2.5. Mejoras en el uso del Dynamic Time Warping

En el apartado 'Estado previo de PerseumAI' 3.1, concretamente en el paso 3 del funcionamiento, se especifica que se seleccionaba el patrón más similar (de los patrones de ejemplo) a la ventana que se está analizando. El objetivo de esto es, más adelante, a la hora de analizar las subventanas, comprobar si hay algún otro patrón con una similitud mejor al patrón ya seleccionado.

El cambio implementado en esta parte ha sido hacer una media de las distancias de la ventana a todos los patrones en lugar de seleccionar la menor distancia.

La justificación de este cambio es la siguiente: Que una ventana tenga una buena similitud con uno de los patrones de ejemplo no implica que sea necesariamente similar al patrón

que se trata de encontrar. Puede ocurrir que sea muy similar a uno de los patrones de ejemplo, pero que se aleje bastante del resto de patrones. Al hacer uso de la media, se seleccionarán las ventanas que mas se asemejen a todos los patrones de ejemplo.

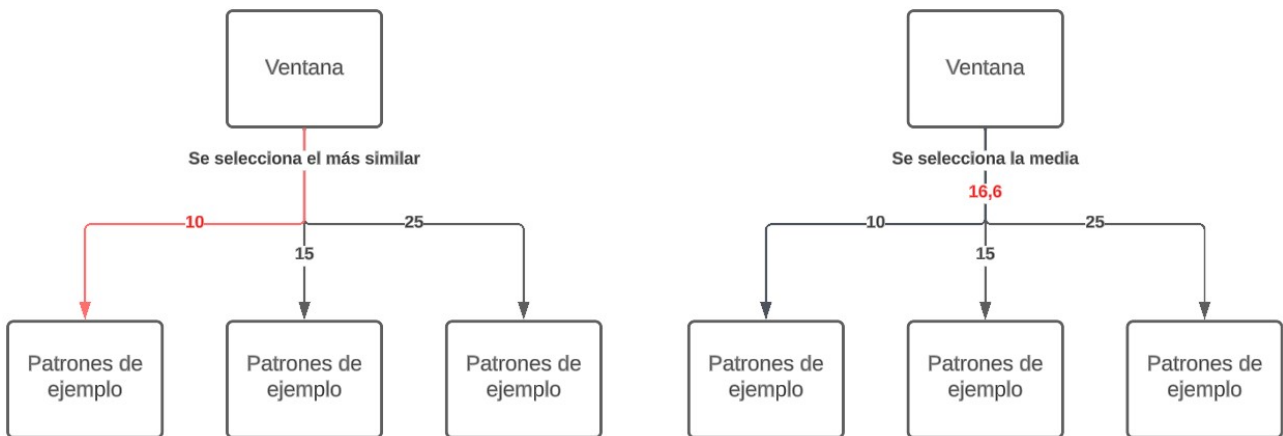


Figura 3.4: Antes vs Después

Cada número de las figuras representa la distancia que hay según el algoritmo Dynamic Time Warping entre la ventana y el patrón de ejemplo.

3.2.6. Mejoras en la detección de patrones individuales

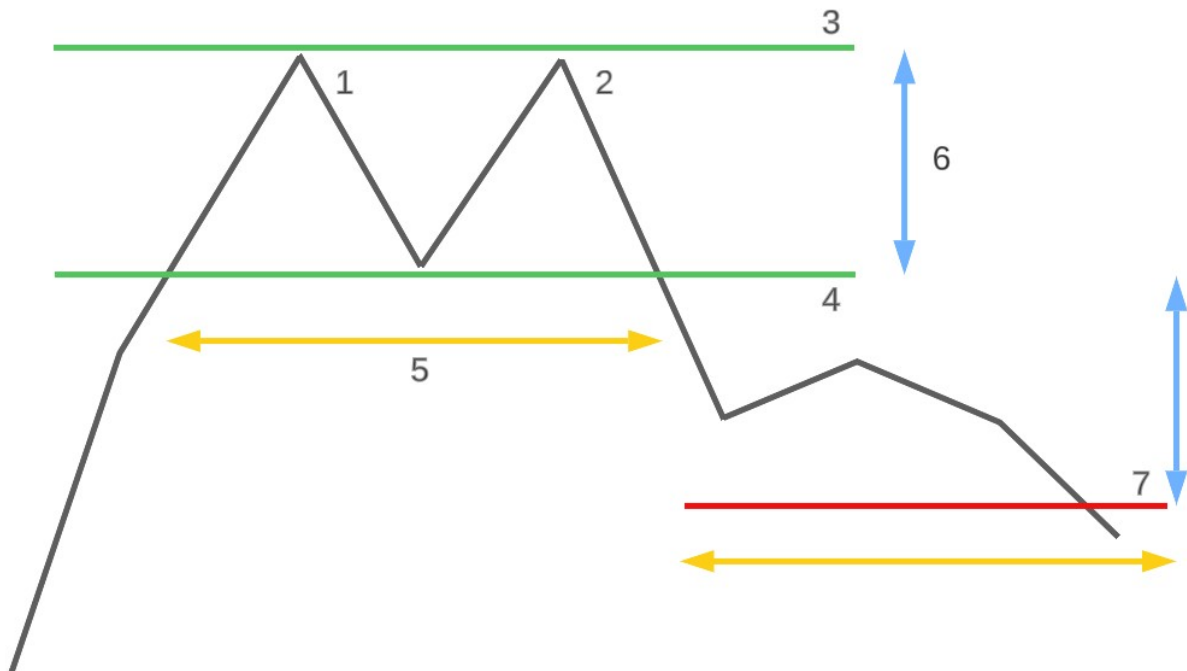


Figura 3.5: Doble techo

En el paso 5 del apartado 'Estado previo de PerseumAI' 3.1 se menciona que la ventana pasa un filtro para comprobar si realmente se parece al patrón seleccionado. Este filtro trata de identificar los dos máximos (3.5, 1 y 2) o los dos mínimos (doble suelo).

Con el objetivo de conseguir que los máximos fuesen mas representativos, se añadió que estos máximos tuviesen que tener dos valores menores tanto por la izquierda como por la derecha, en lugar de uno, ya que a menudo el programa detectaba máximos poco relevantes, ignorando el máximo de mayor importancia que existía en el patrón. Lo mismo se aplicó para los mínimos en el caso del doble suelo.

También se añadió una restricción en la distancia que puede haber entre los dos picos. Normalmente un doble suelo o doble techo no son separados por tan solo 2 o 3 valores, si no que son mas amplios. La distancia mínima entre dos máximos/minimos seleccionada para que pueda considerarse doble techo/suelo se fijó a 5.

Por último se añadieron cambios a como se detectaba si los objetivos de patrón han sido cumplidos. En este caso la modificación fue una simplificación del código, y la eliminación de una restricción la cual hacía que si, por ejemplo, en un doble techo, después de la formación de patrón, se detectaba que el precio de cierre en algún momento subía por encima del cuello (3.5, 4), ese patrón se detectaba como falso (No se cumple). También existía otro problema, en el cual, si después de la formación del patrón, el precio no llegaba a alcanzar su objetivo (3.5, 7), pero tampoco subía por encima del valor del cuello, ese patrón era descartado, ya que no se tuvo en cuenta la posibilidad de que el precio se quedara entre el objetivo y el cuello del patrón. Este problema también fue solucionado, simplificando el código, de manera que, si alguno de los valores de después del patrón alcanzaba el objetivo, se considerará que el patrón se ha cumplido, y en caso contrario, no se habrá cumplido.

3.2.7. Mejoras en la interfaz

Se implementaron algunas mejoras para mejorar la usabilidad de la interfaz, dando al usuario mas control sobre la herramienta, e incluyendo algunas explicaciones o sugerencias en la interfaz para dejar más claro su uso. También se eliminaron algunos elementos que podían llevar a confusión, para tratar de dejar la interfaz mas limpia.

La librería utilizada para la interfaz (tkinter) es una librería que, si bien permite hacer interfaces sencillas para un programa de Python, no permite muchas opciones ni personalización. Más bien, esta librería está pensada para interfaces sencillas. Una alternativa a esta librería habría sido 'pyqt5' la cual permite la creación de interfaces más complejas y personalizables.

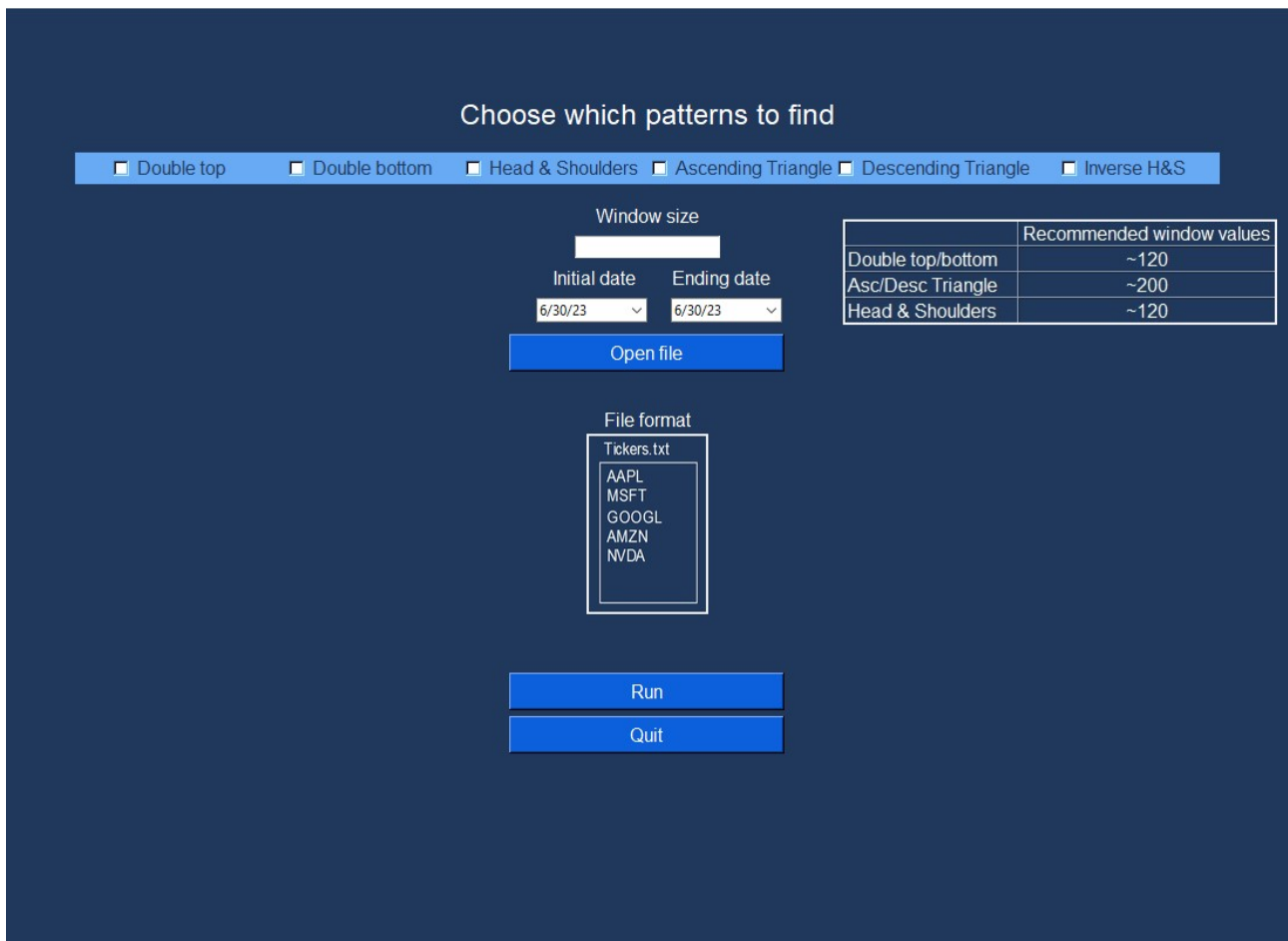


Figura 3.6: Nueva interfaz

Como se ve en la figura, en la nueva interfaz, a parte de haber mas patrones disponibles, en los cuales se profundizará más adelante, también existe la posibilidad de que el usuario escoja el tamaño de ventana. El tamaño de ventana es el tamaño en el que se dividirán los datos para ser analizados (Está explicado con un ejemplo en el apartado 'Estado previo PerseumAI' 3.1). Hay una tabla de recomendaciones a la derecha, con los valores que mejores resultados me han dado a lo largo del desarrollo de la herramienta. Estos valores son aproximados, y siempre se puede experimentar con valores diferentes, para tratar de obtener resultados mejores. La única limitación que tiene el tamaño de ventana es que el valor tiene que ser mayor o igual que 80, ya que internamente el programa divide la ventana hasta en 4 partes, para tratar de encontrar patrones mas claros dentro de si mismo. Al dividirse entre 4, la ventana resultante es de longitud 20, lo que considero que es el menor valor en el que se puede encontrar un patrón.

3.2.8. Cambios en la detección de patrones actuales.

Se rehizo la parte de detección de patrones actuales de una manera diferente. Previamente el código podía funcionar de dos maneras diferentes. La primera era que simplemente se escogía el primer patrón que encontraba desde la fecha '2022-01-01' en adelante. La segunda manera era activando una opción que tenía la antigua interfaz (3.1, 4) de 'Intensive Search Mode', que sí que buscaba entre diferentes patrones y se quedaba con el mejor, pero esta consumía demasiado tiempo debido a la manera en la que lo hacía.

El cambio en este caso vino por hacer que los patrones actuales se detecten de la misma manera que los históricos, pero seleccionando los datos de los últimos X días. Por ejemplo, si la herramienta se ejecuta el día 14/07/2023, se descargarán los datos de los últimos X días, siendo la X el valor de ventana que haya seleccionado el usuario. Una vez descargados esos datos, se analizará esa única ventana en busca de patrones. Esto permite simplificar la búsqueda de patrones, y hacerlo incluso más rápido, pero a cambio de esto, es mucho menos común encontrar patrones actuales.

3.3. Nuevos patrones implementados

En este apartado se abarcarán los nuevos patrones que se han implementado en la herramienta. Para implementar estos patrones, se ha tenido que obtener los datos de ejemplos de estos patrones. Estos nuevos conjuntos de datos han sido obtenidos de la misma manera que los obtuvo Gabriel en su trabajo el año pasado, usando un pequeño programa en el que el usuario puede introducir el nombre de la empresa, la fecha de inicio y final del patrón, y este programa crea el dataframe de manera automática y lo guarda en los archivos. Para detectar que patrones relevantes existen en la actualidad se ha usado la herramienta FinViz, que proporciona información sobre todo tipo de patrones que se están formando en la actualidad, de manera que se usó esta herramienta para consultar que empresas estaban formando patrones en la actualidad, y así después descargar los datos de esas fechas. Una vez descargados los patrones, se desarrolló para cada patrón la función cuyas dos funcionalidades principales son las siguientes:

- Filtro que comprueba si los datos seleccionados (La ventana) cumple los requisitos del patrón. Por ejemplo, comprobar que, en el hombro cabeza hombro, tiene 3 picos, de los cuales el del medio es el mayor, entre otras restricciones en las que profundizaré en el apartado de cada patrón.
- Comprobación de si el objetivo de patrón se ha cumplido. Se comprueban los datos después del patrón detectado, y se traza un objetivo en un plazo de tiempo de la misma longitud que el patrón para comprobar si el patrón ha logrado cumplir su objetivo.

3.3.1. Hombro cabeza hombro

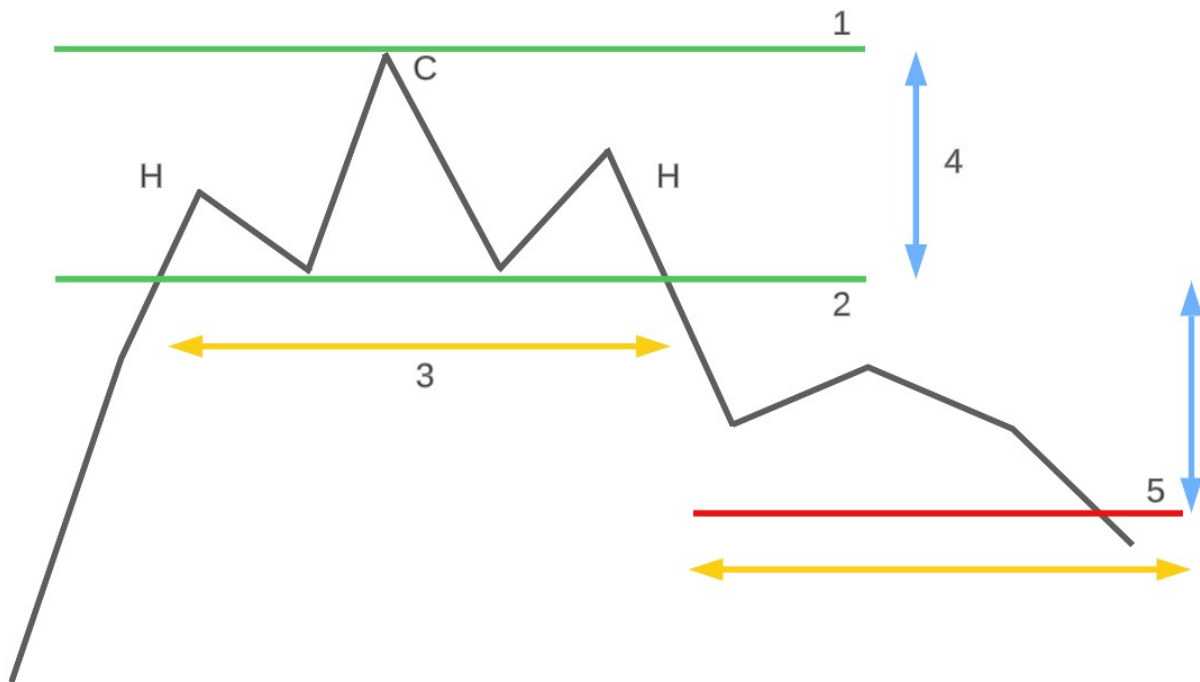


Figura 3.7: Hombro cabeza hombro componentes

El hombro cabeza hombro fue el primer patrón implementado durante el desarrollo. El objetivo de este patrón se traza de la misma manera que el doble techo 3.5, lo cual se puede ver en la figura anterior. El objetivo se traza calculando la distancia entre la cabeza (3.7, C) y el cuello (3.7, 2), y luego restando esa distancia al cuello, trazando así el objetivo (3.7, 5). Se considera que el hombro cabeza hombro ha cumplido su objetivo si lo alcanza en un plazo de tiempo similar al de la longitud del patrón (3.7, 3).

A continuación se mostrarán dos patrones de este tipo encontrados por la herramienta, uno que ha cumplido su objetivo, y otro que no lo ha hecho.

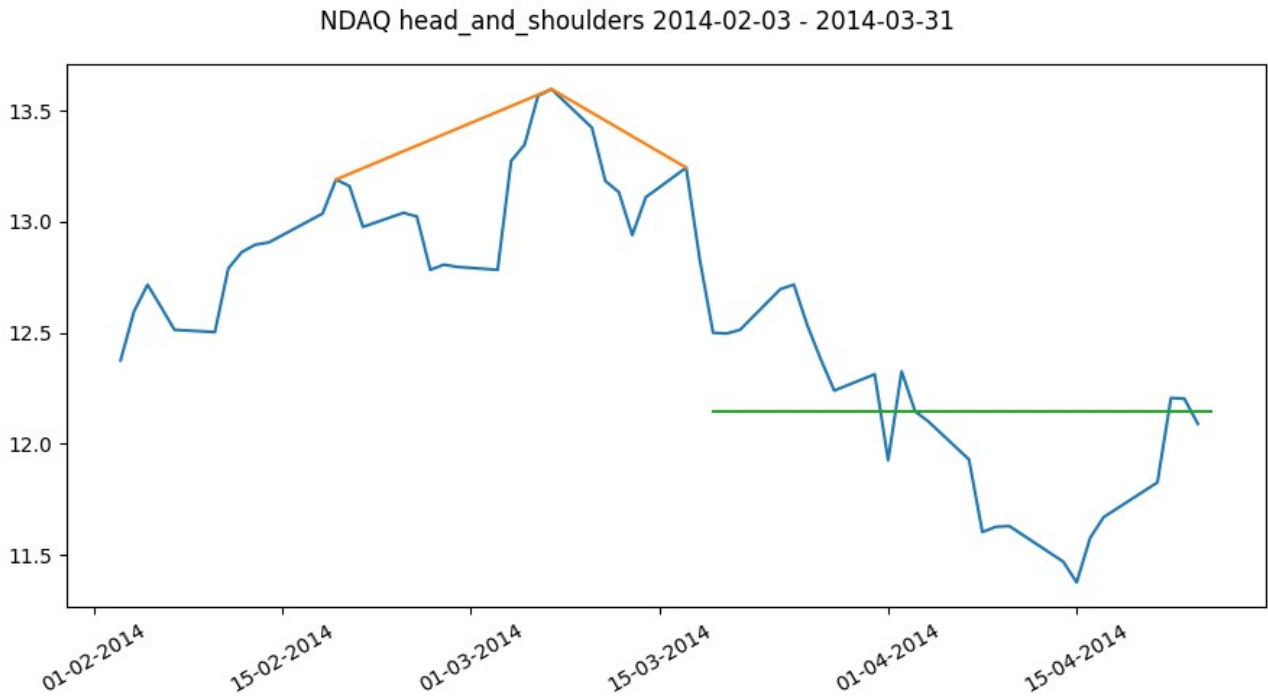


Figura 3.8: HCH con el objetivo cumplido

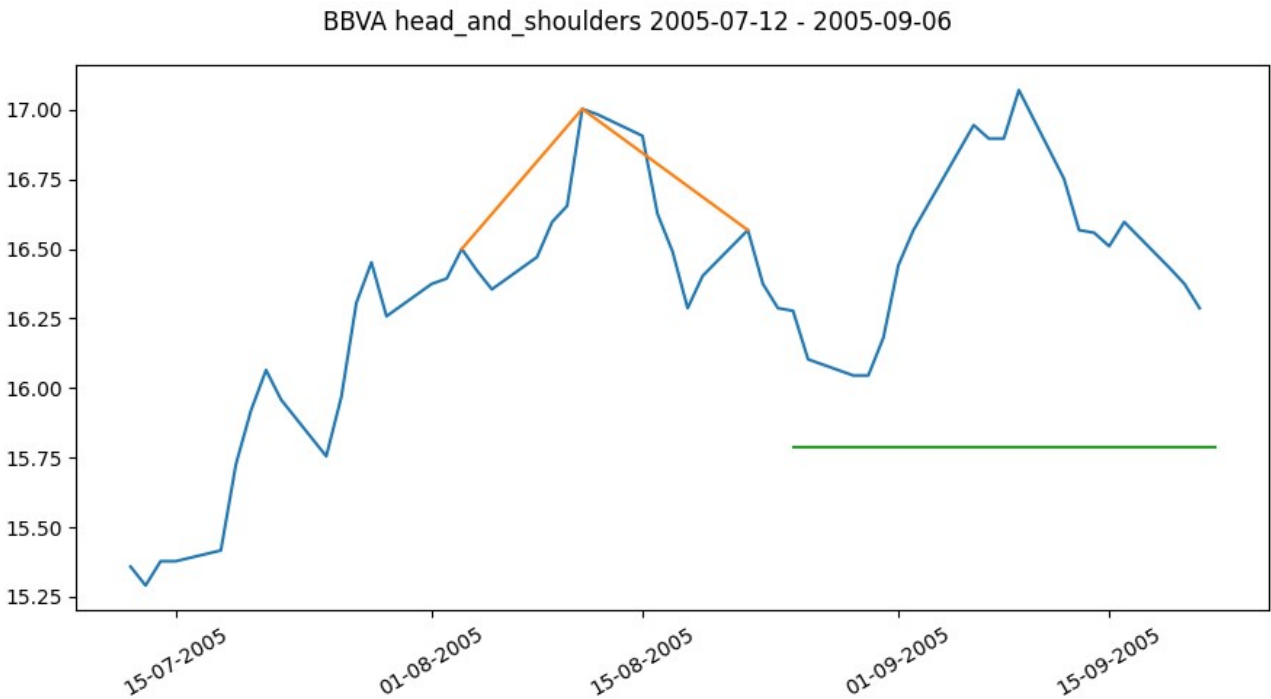


Figura 3.9: HCH con el objetivo no cumplido

Como se mencionó en el apartado 3.3 cada patrón tiene un filtro que comprueba si, después de pasar por el filtro de distancia del Dynamic Time Warping, este cumple con las condiciones del patrón. En el caso del hombro cabeza hombro, lo más importante es poder identificar los 3 máximos relativos que lo componen. Para ello, se recorre el patrón comparando todos los puntos con sus dos puntos contiguos por el lado izquierdo, y sus dos puntos contiguos por el lado derecho. Si el punto seleccionado es mayor que los dos puntos de cada lado, se le considerará un máximo relativo. Una vez encontrados

e identificados los 3 máximos relativos, se pasará a hacer una serie de comprobaciones entre estos 3 puntos para comprobar si cumple con las condiciones del patrón. Estas comprobaciones son:

- El máximo más alto de los 3 está ubicado en medio.
- El máximo de en medio es considerablemente mayor que los otros dos.
- Los máximos laterales son similares, y que sus cuellos también lo son.
- El patrón tiene una longitud mínima de 12 puntos.

Una vez el patrón seleccionado haya pasado todas estas condiciones, será confirmado como un patrón hombro cabeza hombro, y se pasará a recorrer la longitud necesaria posterior al patrón para comprobar si el patrón se cumple o no.

3.3.2. Hombro cabeza hombro invertido

Este patrón es igual al anterior pero a la inversa. Igual que el hombro cabeza hombro convencional busca identificar un cambio de tendencia alcista a tendencia bajista, este busca lo contrario, por lo que el objetivo cuando se identifique este tipo de patrón, no será un punto más bajo, si no uno más alto. A continuación se mostrarán las figuras que identifica la herramienta de este tipo de patrón.

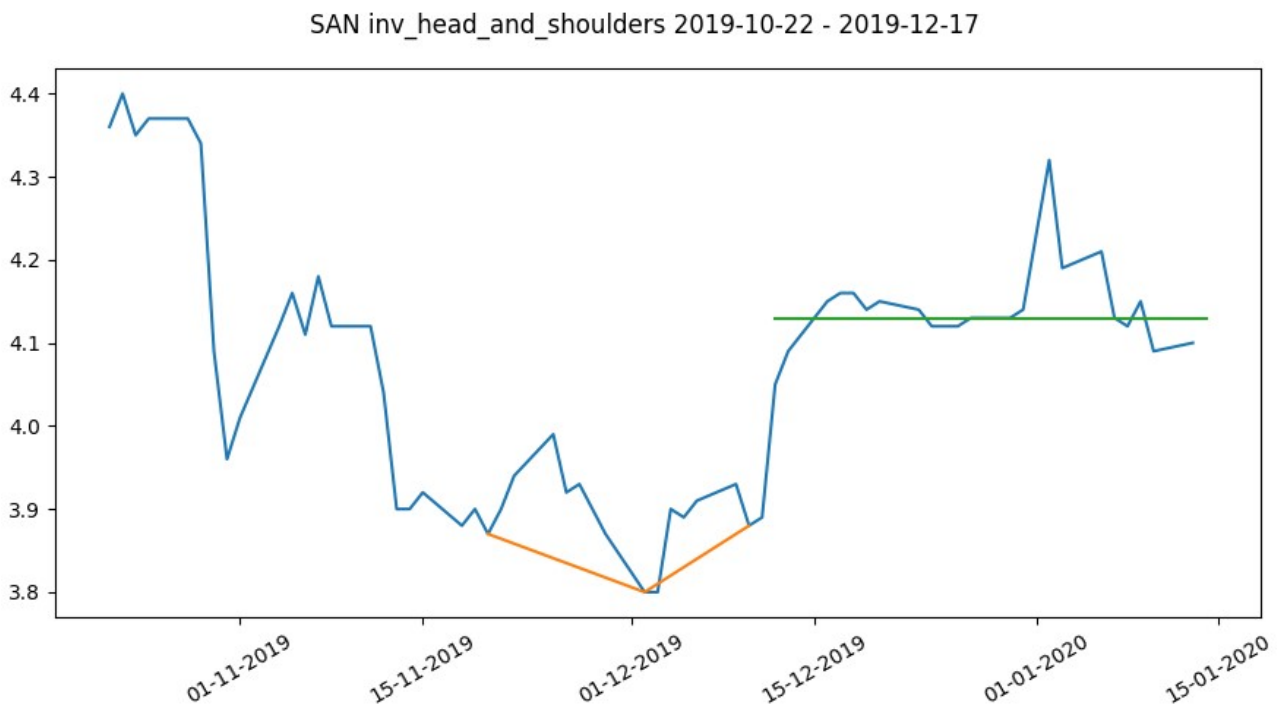


Figura 3.10: HCH invertido con el objetivo cumplido

FCF inv_head_and_shoulders 2020-06-11 - 2020-09-03

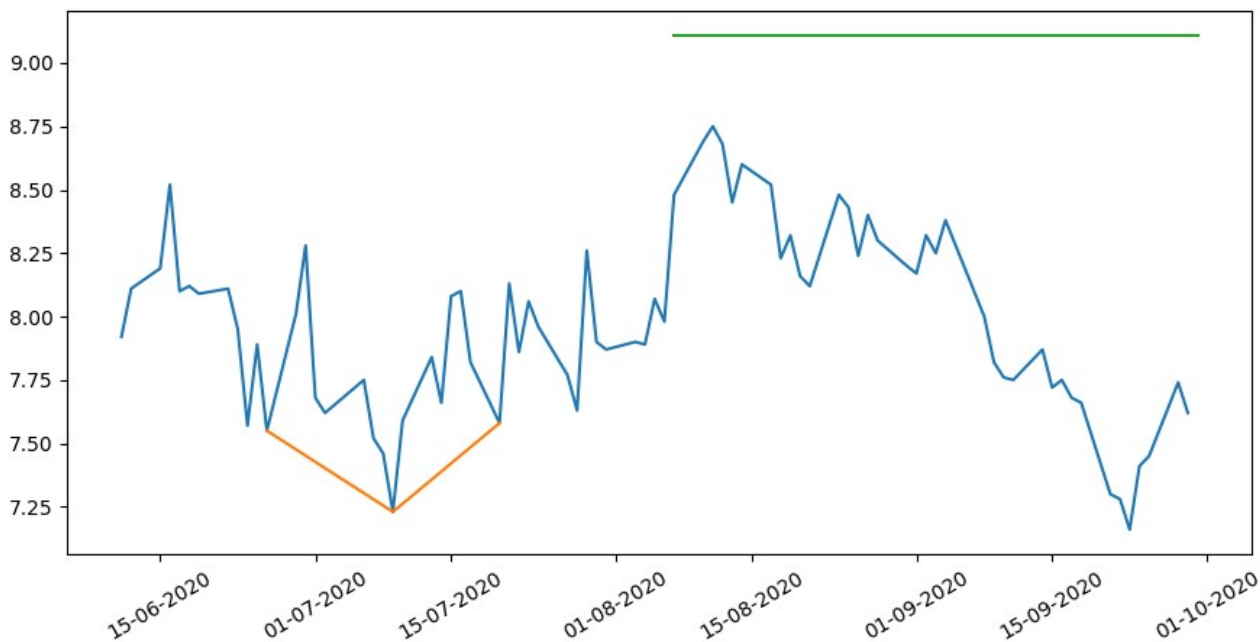


Figura 3.11: HCH invertido con el objetivo no cumplido

Los filtros del patrón hombro cabeza hombro invertido, son los mismos que el de patrón Hombro Cabeza Hombro convencional, pero a la inversa. En este caso se trata de encontrar 3 mínimos relativos. El objetivo del patrón en este caso es, comprobar si el precio sube hasta la suma de la línea de resistencia más la distancia entre la resistencia y el mínimo.

3.3.3. Triángulo ascendente

A diferencia del patrón hombro cabeza hombro, los patrones triángulo ascendente/descendente son patrones de continuación de tendencia, lo que significa que presuntamente confirman que el precio de un activo va a permanecer en una tendencia alcista/bajista.

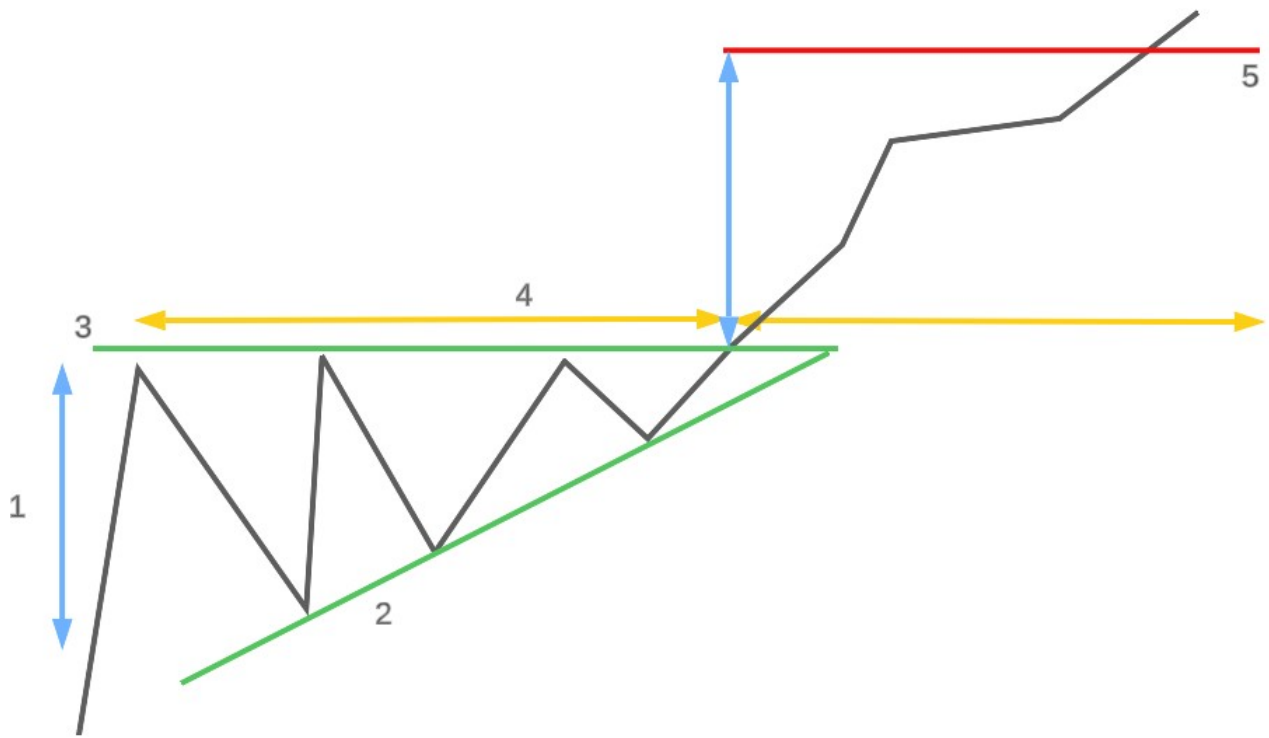


Figura 3.12: Triángulo ascendente componentes

Este patrón define una tendencia alcista cuyos puntos de soporte son cada vez más altos (3.12, 2), y a su vez el precio se encuentra varias veces con una misma resistencia (3.12, 3), la cual acaba por romper eventualmente, para alcanzar así su objetivo (3.12, 5), el cual se traza sumando la altura del triángulo (3.12, 1) más la resistencia, tal y como se ve en la figura.

A continuación se mostrarán las figuras de este tipo que identifican la herramienta, una con el objetivo cumplido, y otra sin cumplirlo.

BLK ascending_triangle 2012-09-24 - 2012-11-06

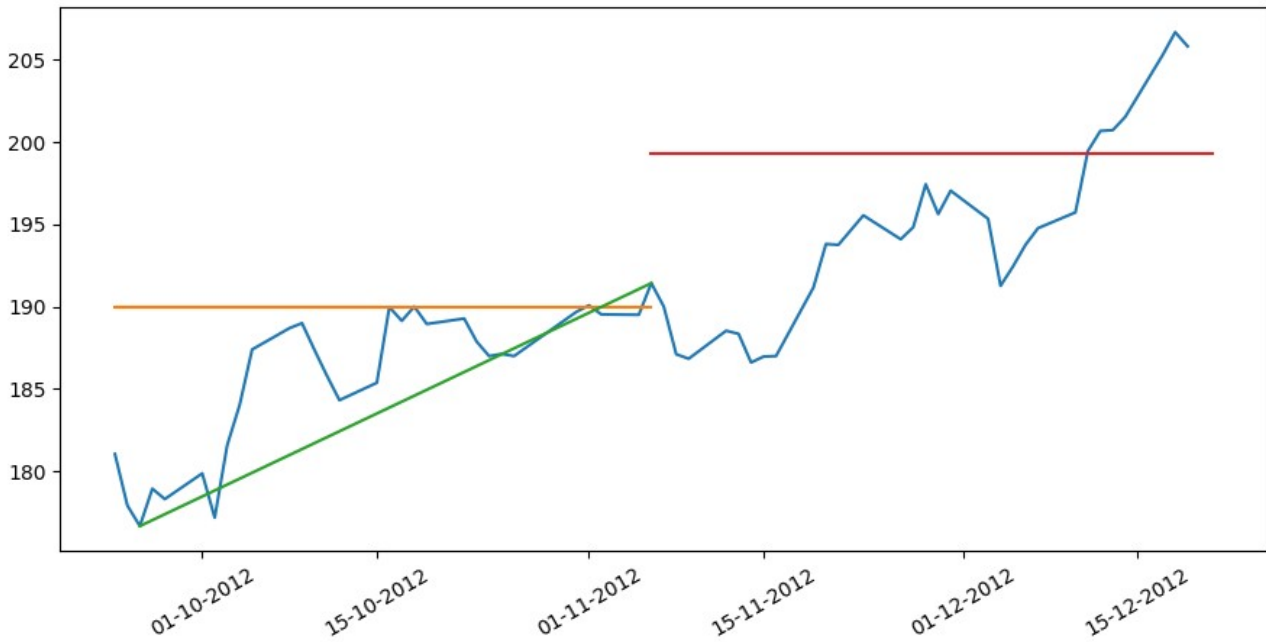


Figura 3.13: Triangulo ascendente con objetivo cumplido

BBAR ascending_triangle 2012-11-29 - 2013-01-11

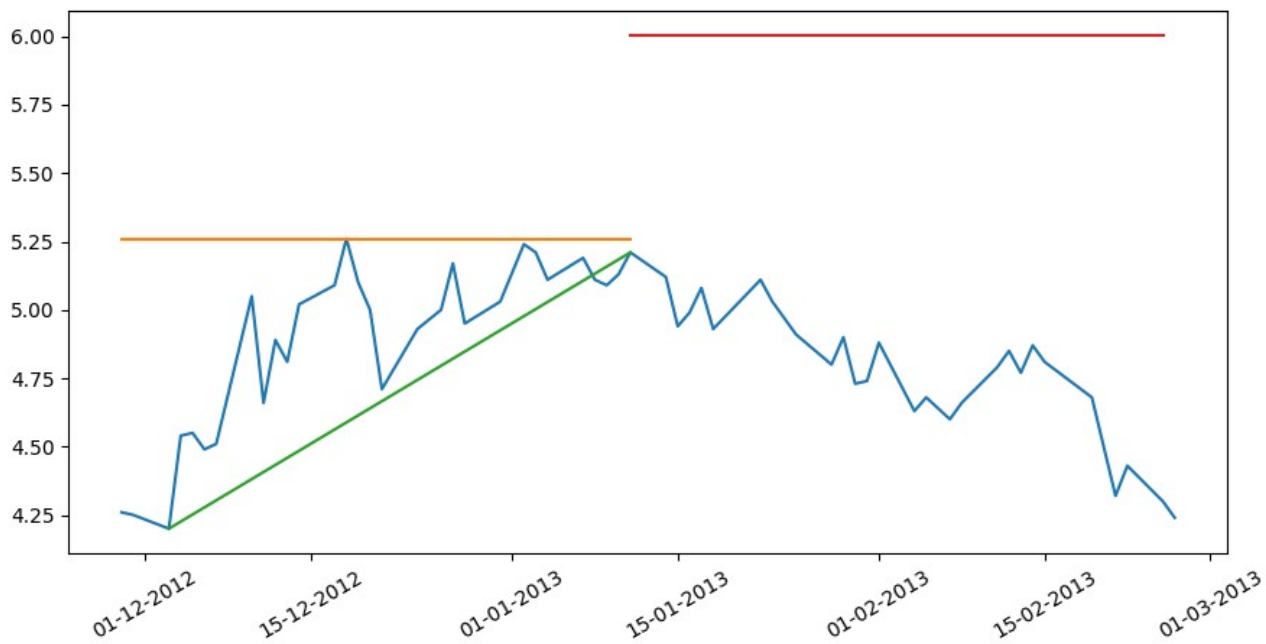


Figura 3.14: Triangulo ascendente con objetivo no cumplido

En este patrón, como los máximos locales no son tan importantes, las comprobaciones han sido mas orientadas a que el patrón mantenga la forma característica del triángulo ascendente. Estas comprobaciones han sido las siguientes:

- El patrón rebota al menos 3 veces en la línea de resistencia.
- Los mínimos relativos son ascendentes.

- El triangulo se ha cerrado correctamente.
- Los valores no se pasan demasiado de la diagonal por debajo.

Una vez se haya comprobado que el patrón cumple las condiciones anteriores, se confirmará que es un triángulo ascendente, y se pasará a comprobar si cumple su objetivo.

Esta herramienta tiene una pequeña problemática a la hora de identificar este patrón, y es que a veces se parece más a un canal ascendente que a un triángulo ascendente, pero es una problemática compleja de solucionar, ya que ambos son patrones muy similares. La buena noticia es que ambos son patrones que confirman una tendencia alcista, con lo que el objetivo es similar.

3.3.4. Triangulo descendente

Este patrón es similar al triángulo ascendente, al igual que lo es el hombro cabeza hombro inverso al hombro cabeza hombro. En este caso el triángulo descendente confirmaría la continuación de una tendencia bajista. A continuación se mostrarán dos ejemplos de este patrón identificados por la herramienta.

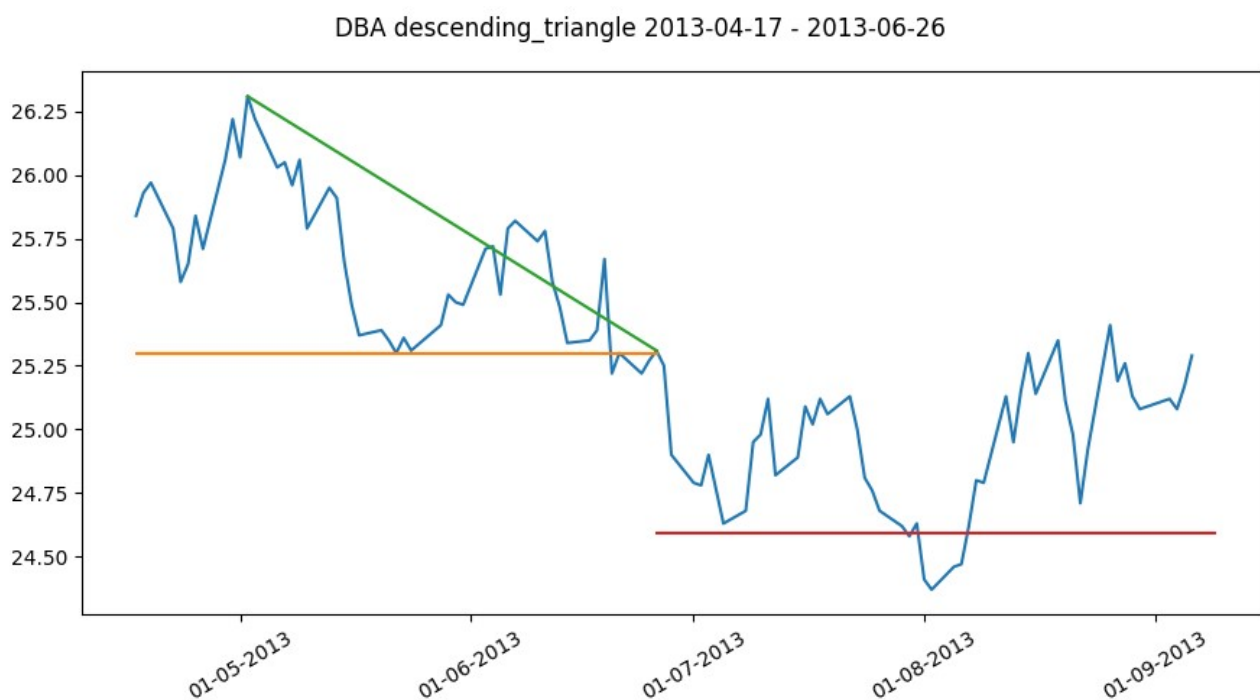


Figura 3.15: Triangulo descendente con objetivo cumplido

BBAR descending_triangle 2006-04-27 - 2006-06-22

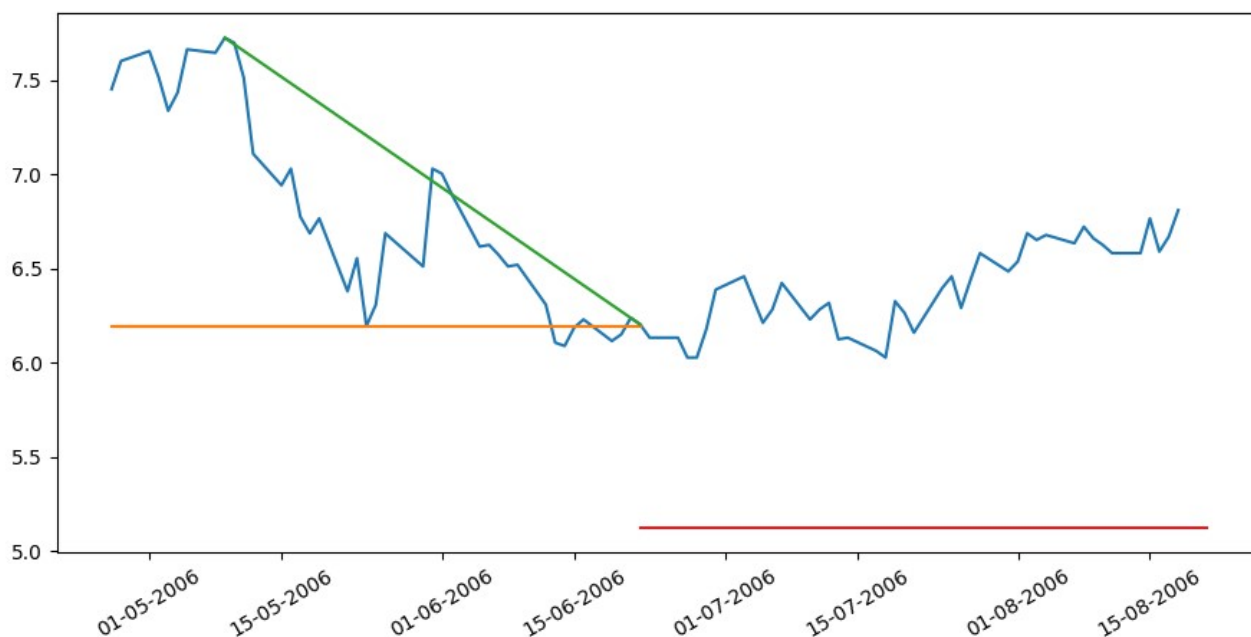


Figura 3.16: Triangulo descendente con objetivo no cumplido

3.4. Distribución de la aplicación

Para la distribución de la aplicación se ha creado un archivo ejecutable mediante el software 'Auto-py-to-exe', el cual permite, mediante una interfaz gráfica, poder indicar todos los parámetros y configuraciones necesarias a la hora de crear el ejecutable. Este archivo ejecutable se encuentra disponible en [Google Drive](#). También se puede acceder a este archivo ejecutable y al resto del código a través del [repositorio](#) creado en Github.

3.5. Algoritmos adicionales

En el repositorio se encuentra una carpeta llamada 'auxScripts' en la cual hay diferentes archivos, que si bien no forman parte del programa principal, son programas que han sido de apoyo para el desarrollo del proyecto. Entre ellos se encuentran archivos como el archivo de la configuración que he usado para la creación del ejecutable mediante 'auto-py-to-exe', por si se quisiera modificar el código y crear un ejecutable nuevo. También en la misma línea se encuentra un archivo llamado 'hook-frozendict.py' que soluciona un problema con la creación de ese ejecutable.

Por otro lado hay un script llamado 'seedata.py' que permite ver los datos gráficamente del patrón seleccionado. Por ejemplo, se pueden visualizar todos los patrones de ejemplo de doble techo, o todos los de hombro cabeza hombro, haciendo uso de la misma librería de visualización que se usa en el proyecto, 'matplotlib'.

También hay otro script llamado 'table.py', el cual crea la tabla que se ve en el programa original de los valores recomendados de ventana. Luego, mediante una captura de pantalla, se incluye esta tabla en el programa original. Esto se ha realizado de esta manera ya que habían incompatibilidades entre el código de la tabla y el resto de la

interfaz del programa.

El script llamado 'average_length.py' calcula la longitud media de los ejemplos de los patrones. Es decir, cuenta cuantos datos hay en cada fichero, y hace una media entre todos, para cada patrón.

Por último, un fichero llamado 'cross_validation.py' contiene un script que ejecuta la técnica llamada cross-validation en el conjunto de los datos de ejemplos de los patrones. El uso de este fichero está explicado más adelante, en el apartado 4.2.4.

Capítulo 4

Resultados obtenidos

4.1. Desempeño de la herramienta en datos históricos

Con el objetivo de analizar algunos resultados obtenidos por la herramienta, será realizado un pequeño backtesting con la misma, ya que esta es capaz de proporcionar un porcentaje de cumplimiento de los patrones, es decir, que indica cuantos patrones han alcanzado su objetivo. Esto se comprobará para diferentes sectores en diferentes periodos temporales.

4.1.1. Sector tecnológico

El siguiente conjunto de empresas pertenecientes al sector tecnológico fue analizado:

Ticker	Compañía	Cap. de Mercado
AAPL	Apple Inc.	\$3045 B
MSFT	Microsoft	\$2538 B
NVDA	NVIDIA Corporation	\$1052 B
ORCL	Oracle Corporation	\$319 B
CSCO	Cisco Systems, INC	\$209 B
NOK	Nokia Oyj	\$23.83 B
WDC	Western Digital Corporation	\$12.12 B
LOGI	Logitech International S.A.	\$10.27 B
LPL	LG Display Co., Ltd.	\$4.35 B
NEON	Neonode Inc.	\$121.04 M

Tabla 4.1: Conjunto de empresas del sector tecnológico

Las empresas se han seleccionado gracias a la herramienta FinViz, ya que esta permite filtrar por diferentes opciones. Todas las empresas seleccionadas tienen más de 15 años desde su oferta pública inicial, para que haya una buena cantidad de datos a analizar. También se han tratado de seleccionar empresas con diferentes capitalizaciones de mercado, desde empresas tan grandes como 'Apple' y 'Microsoft' a empresas con una capitalización menor como puede ser 'Neonode Inc.'. Las capitalizaciones de mercado de las tablas son a fecha 5 de Julio de 2023. El periodo seleccionado para esta prueba ha sido desde el 1 de enero de 2005 hasta el 3 de julio de 2023. Los tamaños de ventana seleccionados han sido 120 para dobles techo, dobles suelo, hombro cabeza hombro y

hombro cabeza hombro invertido, mientras que para los triángulos ascendente y descendente se ha seleccionado un tamaño de ventana de 200. También han sido seleccionados todos los patrones existentes en la herramienta, para así hacer un análisis completo de estas compañías en esta época. Los resultados han sido los siguientes:

Patrón	%	Nº
Doble techo	67 %	83
Doble suelo	69 %	81
Hombro cabeza hombro	73 %	19
Hombro cabeza hombro invertido	72 %	36
Triángulo ascendente	82 %	11
Triángulo descendente	50 %	2

Tabla 4.2: Resultados de 'PerseumAI' de las empresas del sector tecnológico (A.3)

En la tabla anterior, la columna '%' representa el porcentaje de patrones del total que han cumplido con el objetivo, y la columna 'Nº' representa el número total de patrones encontrados. Por ejemplo, en el doble techo se puede ver que la herramienta ha encontrado 39 patrones, de los cuales el 59 % (23 patrones) han llegado a cumplir el objetivo.

4.1.2. Sector financiero

Para el sector financiero se ha utilizado el siguiente conjunto de datos:

Ticker	Compañía	Cap. de Mercado
V	Visa Inc.	\$471 B
JPM	JPMorgan Chase & Co.	\$407 B
BLK	BlackRock, Inc.	\$102 B
SAN	Banco Santander, S.A.	\$58 B
BBVA	Banco Bilbao Vizcaya Argentaria, S.A.	\$44 B
BBD	Banco Bradesco S.A.	\$33.56 B
NDAQ	Nasdaq, Inc.	\$23.90 B
DB	Deutsche Bank Aktiengesellschaft	\$20.67 B
BBAR	Banco BBVA Argnetina S.A.	\$2.3 B
FCF	First Commonwealth Financial Corporation	\$1.29 B

Tabla 4.3: Conjunto de empresas del sector financiero

Este conjunto de empresas ha sido seleccionado de la misma manera que el anterior, con más de 15 años de datos, y tratando de seleccionar diferentes capitalizaciones de mercado a día 5 de Julio de 2023. También se ha seleccionado la fecha desde el 1 de enero de 2015 hasta el 3 de julio de 2023, y los mismos tamaños de ventana. Estos han sido los resultados:

Patrón	%	Nº
Doble techo	58 %	79
Doble suelo	56 %	79
Hombro cabeza hombro	47 %	15
Hombro cabeza hombro invertido	77 %	35
Triángulo ascendente	36 %	11
Triángulo descendente	25 %	4

Tabla 4.4: Resultados de 'PerseumAI' de las empresas del sector financiero (A.1)

4.1.3. Análisis de resultados

Dado que este es un trabajo dedicado a la herramienta de detección de patrones, y no a un análisis profundo de los resultados, simplemente se mencionaran algunos puntos llamativos acerca de los resultados. Lo más notorio es que los doble techo/suelos son mucho más abundantes que el resto de patrones, dado que es un patrón que es bastante simple, y fácil de identificar en los datos. Otra cosa a mencionar es que, dado que el número de patrones encontrados es relativamente pequeño, es muy fácil encontrar que hay porcentajes muy altos o muy bajos, que difícilmente se podrían extrapolar a otro tipo de casos. Además, de que rentabilidades pasadas no garantizan rentabilidades futuras.

4.2. Selección de parámetros

Para que el programa funcione hace uso de distintos parámetros, unos seleccionados por el usuario, y otros que han sido fijados en el código, pero que sería interesante tener en cuenta ya que se podrían variar.

4.2.1. Tamaño de ventana

En la interfaz del programa se proporcionan unos tamaños de ventana recomendados, pero ¿por qué esos valores? La realidad es que ha sido una mezcla entre experiencia a la hora de desarrollar la herramienta, y los tamaños existentes de los patrones de ejemplo. Los patrones de ejemplo han sido obtenidos de FinViz, una herramienta que proporciona información sobre patrones en formación.

4.2.2. Longitud de media móvil

Otro parámetro, que en este caso no es controlado por el usuario, si no está incluido en el programa, es la longitud de la media móvil. Una media móvil más larga proporciona una suavización más alta, pero también puede introducir un retraso en la detección de cambios o tendencias en los datos. Es importante tener en cuenta que al aplicar una media móvil, se está introduciendo un cierto grado de suavizado y pérdida de información temporal. Esto puede afectar la precisión de la comparación con DTW, ya que algunos detalles finos de las series temporales pueden perderse. Como en este proyecto se pretende detectar movimientos clave, que pueden suceder en muy pocos datos, se ha elegido una longitud de **3**. Esta longitud permite quitar algo de ruido, pero sin perder el detalle de los movimientos realizados por el precio.

4.2.3. Cantidad de datos de ejemplo seleccionados

Uno de los parámetros que también está incluido en el código es la cantidad de conjuntos de datos de ejemplo que se seleccionan del total del conjunto de datos de ejemplo existentes. Por ejemplo, tanto en el doble techo como en el doble suelo existen 50 ejemplos de ellos, pero el programa cada vez que se ejecuta solo utiliza 15 de esos ejemplos (Seleccionados de manera aleatoria). Esto se hace ya que seleccionarlos todos haría que el programa tardara una cantidad demasiado larga de tiempo para ejecutarse, ya que Dynamic Time Warping es un algoritmo que da buenos resultados, pero no es muy eficiente. Esto se puede verificar en artículos como (Sharabiani et al, 2017), en el cual, argumentando que el algoritmo, si bien es el algoritmo más utilizado para clasificar series temporales, no es muy eficiente dada su complejidad cuadrática.

Cada vez que una ventana es analizada, el programa tiene que comparar esa ventana con cada uno de los 15 ejemplos de cada patrón seleccionado. Esto quiere decir, que si se selecciona por ejemplo, doble techo y hombro cabeza hombro, el programa ha de realizar 30 comparaciones por cada ventana. Si se ha seleccionado 1 sola empresa, durante un periodo de 10 años ($365 * 10 = 3650$ datos) con un tamaño de ventana de 120, en el mejor de los casos, esos 3650 datos se dividirán en $3650 / 120 = 31$ ventanas, generando así $31 * 30 = 930$ comparaciones utilizando el Dynamic Time Warping. Dado que el programa no solo busca en la ventana seleccionada, si no que luego trata de buscar en 9 subventanas dentro de si mismo, tal y como mencioné en el apartado 3.1, tiene que comparar cada una de estas 9 subventanas con los 30 patrones ($9 * 30 = 270$). Esto añadiría un total de 270 comparaciones por cada ventana. $(31 * 30) + (270 * 31) = 9300$ comparaciones.

Como se ve con el ejemplo anterior, se realizan demasiadas comparaciones mediante este algoritmo, razón por la que el programa es un poco lento, y añadir más patrones de ejemplo aumentaría exponencialmente las comparaciones y por lo tanto, el tiempo de ejecución del programa.

4.2.4. Distancia para filtrar

Otro de los parámetros del programa es la distancia máxima que puede dar como resultado la comparación de Dynamic Time Warping. Como se ha mencionado en el apartado 3.2.3, una de las mejoras añadidas es descartar patrones cuya distancia fuera considerablemente alta.

Este valor fue seleccionado considerando las diferentes distancias obtenidas al aplicar el método de '**cross-validation**'. En este método, a pesar de que normalmente se usa en machine learning, consiste en dividir el conjunto de datos de ejemplo en dos subconjuntos, uno para 'entrenamiento' y otro para 'prueba'. En este caso se ha utilizado el conjunto de 'entrenamiento' como los ejemplos y el conjunto de 'prueba' como potenciales entradas de datos. Esto se ha realizado con el objetivo de comprobar cuanta distancia hay en los ejemplos de patrones entre si mismos. Después de ejecutar el programa 10 veces para cada patrón, ha dado, de media, los siguientes resultados, mostrados en la tabla 4.5

Patrón	Distancia
Doble techo	4.5
Doble suelo	3.2
Hombro cabeza hombro	13.4
Hombro cabeza hombro invertido	18.1
Triangulo ascendente	12.6
Triangulo descendente	12.4

Tabla 4.5: Distancias medias según DTW de los ejemplos de los patrones entre sí

En estos resultados se puede observar que la distancia entre los conjuntos de datos de un mismo patrón entre si, varían según el patrón. Analizando esta distancias, teniendo en cuenta las diferencias entre patrones, y también los diferentes tamaños de ventana que se puedan seleccionar, se decidió poner un filtro que descartara patrones con una distancia mayor a **40**.

4.3. Análisis de errores, limitaciones y posibles mejoras

4.3.1. Doble techo VS hombro cabeza hombro

Una problemática que surgió durante el desarrollo es que, al seleccionar a la vez el patrón hombro cabeza hombro junto con el doble techo, este último hacía que apenas hubieran patrones hombro cabeza hombro. La razón de este problema es que, cuando una ventana de datos es comparada con los datos de ejemplo, esta es comparada con todos los ejemplos de los patrones seleccionados a la vez.

Para explicarlo con un ejemplo, imaginemos que se está analizando una ventana para comprobar si existe algún patrón en ella, en este caso, se ha seleccionado el hombro cabeza hombro y el doble techo. La herramienta compara mediante Dynamic Time Warping, la ventana seleccionada (o datos de entrada) con los 15 patrones de doble techo, y con los 15 patrones de hombro cabeza hombro. Cada comparación dará como resultado una distancia, la distancia entre la ventana y cada uno de los 30 patrones (15 de doble techo, y 15 de hombro cabeza hombro). Se hará una media con las 15 distancias de cada patrón, dando una distancia media de X_1 para hombro cabeza hombro y una distancia media de X_2 para doble techo. La herramienta escogerá la media con la distancia mas baja, ya que será el patrón al que mas se asemeja. El problema es que el hombro cabeza hombro y el doble techo son relativamente similares, y es muy fácil encontrar dobles techo en patrones que claramente son hombros cabeza hombro, y al ser el doble techo un patrón mas simple, sus distancias tienden a ser menores que las del hombro cabeza hombro. Pasa exactamente la misma problemática con el doble suelo y el hombro cabeza hombro invertido.

Se puede ver en la figura 4.1 un patrón similar al hombro cabeza hombro, considerado un doble techo por la herramienta.

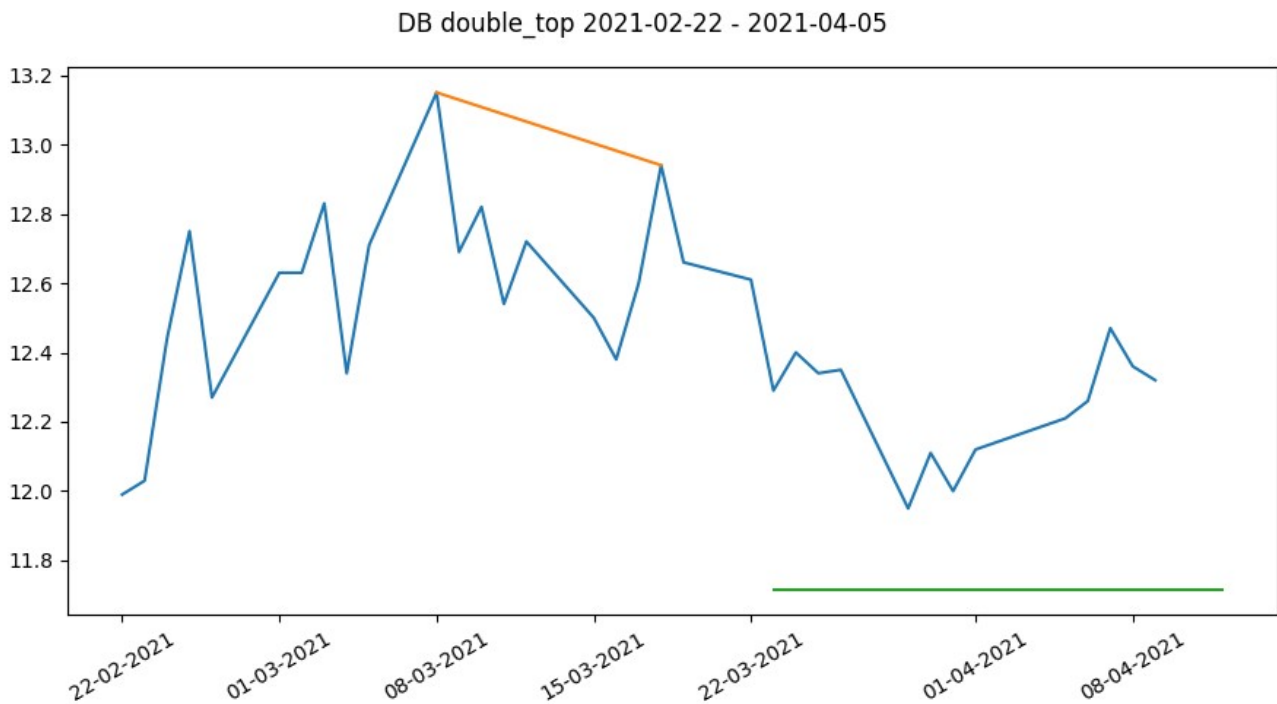


Figura 4.1: Doble techo más similar a hombro cabeza hombro

Solución aplicada: Se ha realizado un cambio para que, cuando estén presentes estos dos patrones a la vez, ya sea doble techo con HCH o doble suelo con HCHI, se ejecuten las comparaciones de manera individual. Es decir, la parte del programa que hace las comparaciones con Dynamic Time Warping, hará primero todas las comparaciones con un solo patrón (Doble techo por ejemplo) y luego en una siguiente iteración las realizará de nuevo, solo con el hombro cabeza hombro, de manera que no comparará las medias entre sí. Con este cambio queda solucionado el problema.

4.3.2. Triángulos similares a canales o cuñas

Uno de los problemas de la herramienta en cuanto al triángulo ascendente y descendente, es que a menudo se pueden confundir con otros patrones chartistas como el canal descendente/ascendente y la cuña ascendente/descendente.

En el caso del canal descendente, no es muy relevante, ya que también es un patrón de continuación, es decir, un canal descendente, predica el mismo objetivo que el triángulo descendente, una continuación de la tendencia. En la figura 4.2 se puede ver un patrón que, a pesar de haber sido detectado como triángulo descendente por la herramienta, es más similar a un canal descendente.

LOGI descending_triangle 2021-06-03 - 2021-09-03

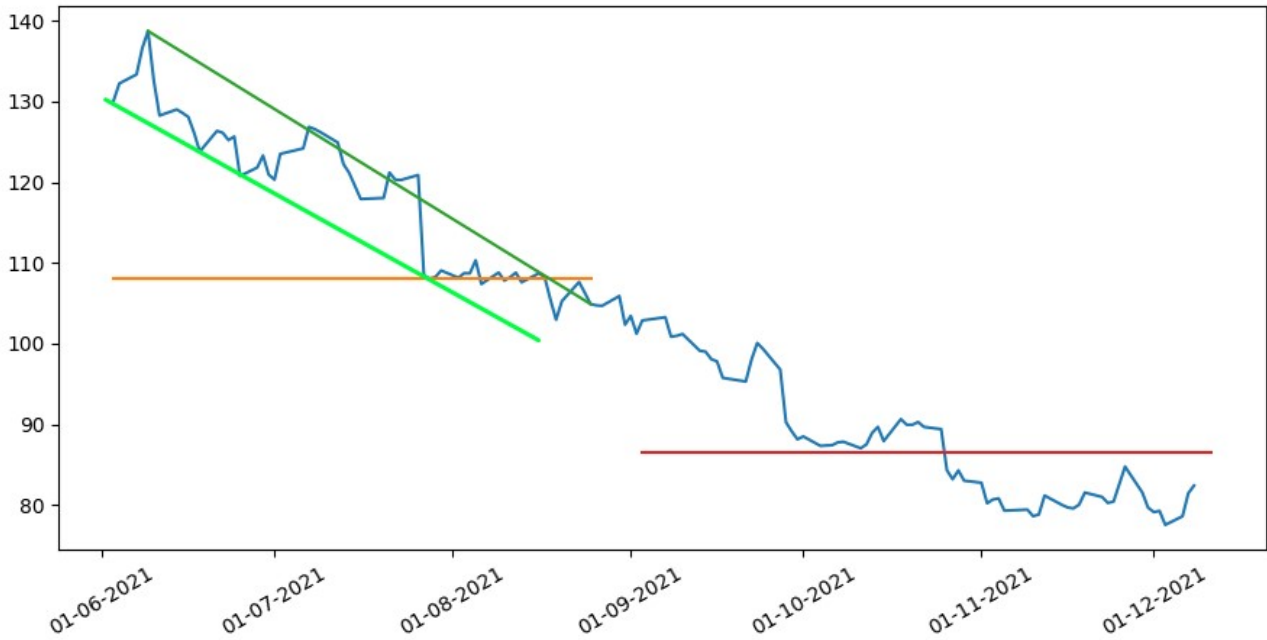


Figura 4.2: Triangulo descendente ambiguo

En cambio, cuando se trata de la cuña, es mas problemático, ya que la cuña, dependiendo del tipo de cuña (ascendente o descendente) y de la tendencia anterior, esta se puede considerar patrón de continuación o de reversión.

Para este problema no se ha encontrado solución, ya que ambas figuras son bastante similares, y es complicado discernir un tipo de patrón del otro de manera fiable.

Capítulo 5

Conclusiones y líneas futuras

5.1. Conclusiones

Este trabajo de fin de grado representa un avance en el campo de la detección automatizada de patrones chartistas en el análisis técnico de la bolsa. Al dar continuidad al trabajo de fin de grado del compañero Gabriel, se ha podido realizar múltiples mejoras a la herramienta incluyendo más flexibilidad a la hora de introducir parámetros, aspectos visuales de patrones encontrados, objetivos y fechas. También se han ampliado el tipo de patrones que puede encontrar la herramienta incluyendo en el caso de patrones de cambio de tendencia, el hombro cabeza hombro y el hombro cabeza hombro invertido; y en el caso de patrones de continuación, el triángulo ascendente y descendente.

Por otro lado, aparte de las diferentes mejoras en la eficiencia y precisión de la herramienta, también se ha optimizado la interfaz de usuario, tratando de hacer hincapié en la usabilidad y la visualización clara de resultados. Esto permite a los usuarios utilizar la herramienta de manera más eficiente y aprovechar al máximo sus capacidades.

Es importante destacar que esta herramienta sigue siendo una ayuda en el proceso de análisis y toma de decisiones, y no pretende reemplazar el juicio humano. Los resultados obtenidos deben ser interpretados en conjunto con otras técnicas de análisis, así como con un análisis fundamental de los activos financieros, para una toma de decisiones informada. Esta herramienta permitirá que los analistas técnicos puedan dar con precisión, el porcentaje de acierto de una figura chartista en particular, pudiendo escoger un subconjunto de empresas, por ejemplo, del sector servicios, tecnológico, etc. en un periodo particular, en vez de decir que una figura chartista tiene una probabilidad alta de suceder.

En resumen, este trabajo representa la continuación de un valioso trabajo de fin de grado previo, proporcionando algunas mejoras significativas en la herramienta. Este trabajo puede resultar interesante ya que, como se ha mencionado en los antecedentes, el análisis de patrones chartistas es un campo en el cual hay muy poca investigación, dejando oportunidades claras para continuar con la exploración de este campo.

5.2. Líneas de trabajo futuras

Hay diversas maneras de proseguir con el desarrollo de esta herramienta, para así tratar de expandir o mejorar la funcionalidad de la herramienta.

Lo más intuitivo y sencillo puede simplemente ser la implementación de nuevos patrones a la herramienta. Si bien la herramienta ya cuenta con 6 patrones, todavía quedan muchos patrones en el análisis chartista que pueden ser analizados, como por ejemplo la cuña, canal, banderas etc. Para esto se tendrían que recopilar nuevos conjuntos de datos que representen estos patrones, y posteriormente hacer algunos cambios en el código para poder añadir el patrón deseado, así como desarrollar la manera en la que se filtrará ese patrón, además de la manera en la que se determinará si el patrón ha cumplido su objetivo o no.

Por otro lado, se podría probar a experimentar con los parámetros de la herramienta, para comprobar cuales son los mejores parámetros a elegir, ya que muchos de los parámetros se han seleccionado en base a la experimentación.

Otra posibilidad interesante podría ser incluir el volumen en los cálculos de la herramienta, de manera que sea una variable más en la que apoyar los cálculos, ya que el volumen es una variable muy relevante en el análisis técnico. Este indicador nos proporciona información sobre la cantidad de compras/ventas que han sucedido en un determinado periodo de tiempo, lo cual podría ser una variable interesante a tener en cuenta. Además, la librería utilizada para la descarga de datos 'yfinance', ya proporciona el volumen cuando descarga los datos de las empresas seleccionadas, así que sería cuestión de simplemente incluir los cálculos deseados en la herramienta.

Por otra parte, se podría contemplar la posibilidad de migrar la interfaz de la herramienta de la librería TkInter a otra librería, como por ejemplo PyQt5. Si bien TkInter es una biblioteca gráfica sencilla, que permite la creación de interfaces de una manera muy simple, carece de opciones de personalización, lo cual hace que sea un poco complicado crear interfaces mas complejas. Si se pretende extender el comportamiento de la herramienta, a la vez que añadir variables y parámetros nuevos, sería recomendable tener en cuenta la posibilidad de crear una nueva interfaz para el programa utilizando PyQt5, que si bien es un poco más denso y complejo de aprender, es capaz de generar interfaces mas complejas y profesionales.

Finalmente, otra opción podría ser contemplar la posibilidad de utilizar otros algoritmos derivados del Dynamic Time Warping, que puedan hacer comparaciones más eficientes, como por ejemplo FastDTW.

Estas son sólo algunas de las posibles mejoras para la herramienta, pero seguramente haya un gran espacio de posibilidades, ya sea tanto en la extensión del comportamiento, como en la precisión a la hora de encontrar los diferentes patrones chartistas.

Capítulo 6

Summary and Conclusions

6.1. Conclusions

This bachelor's thesis represents an advancement in the field of automated chart pattern detection in technical analysis of the stock market. By building upon the work of my colleague Gabriel's bachelor's thesis, it has been possible to make multiple improvements to the tool, including more flexibility when entering parameters, visual aspects of found patterns, objectives, and dates. The type of patterns that the tool can find have also been expanded, including in the case of trend reversal patterns, the head-shoulder-shoulder and the head-shoulder-shoulder inverted pattern; and in the case of continuation patterns, the ascending and descending triangle.

Furthermore, in addition to various enhancements in tool efficiency and accuracy, the user interface has also been optimized, emphasizing usability and clear result visualization. This allows users to utilize the tool more efficiently and make the most of its capabilities.

It is important to note that this tool remains an aid in the analysis and decision-making process and does not aim to replace human judgment. The obtained results should be interpreted in conjunction with other analysis techniques, as well as fundamental analysis of financial assets, to make informed decisions. This tool will allow technical analysts to accurately give the percentage of success of a particular chartist figure, being able to choose a subset of companies, for example, from the service sector, technology, etc. in a particular period, instead of saying that a chart figure has a high probability of happening.

In summary, this work represents the continuation of a valuable previous bachelor's thesis, providing significant improvements to the tool. This work is of interest because, as mentioned in the background, chart pattern analysis is a field with limited research, leaving clear opportunities for further exploration in this area.

Capítulo 7

Presupuesto

7.1. Costes materiales

Tipos	Descripción	Coste
Equipo informático utilizado para el desarrollo	Ordenador, pantalla, periféricos	2000 €

Tabla 7.1: Tabla de costes materiales

7.2. Costes de trabajo

Tipos	Horas	Coste/Hora	Total
Investigación	20	20 €	400 €
Familiarización con el código existente	40	20 €	800 €
Implementación de nuevos patrones	45	35 €	1575 €
Implementación de mejoras	40	30 €	1200 €
Mejoras en la interfaz	15	30 €	450 €
Desarrollo de la memoria	40	25 €	1000 €
Total	200	~26.6 €	5425 €

Tabla 7.2: Tabla de costes de trabajo

Apéndice A

Elementos adicionales

A.1. Enlaces al programa

[Repositorio Github \(Código\)](#)

[Google Drive \(Aplicación\)](#)

A.2. Fuentes de datos de las tablas

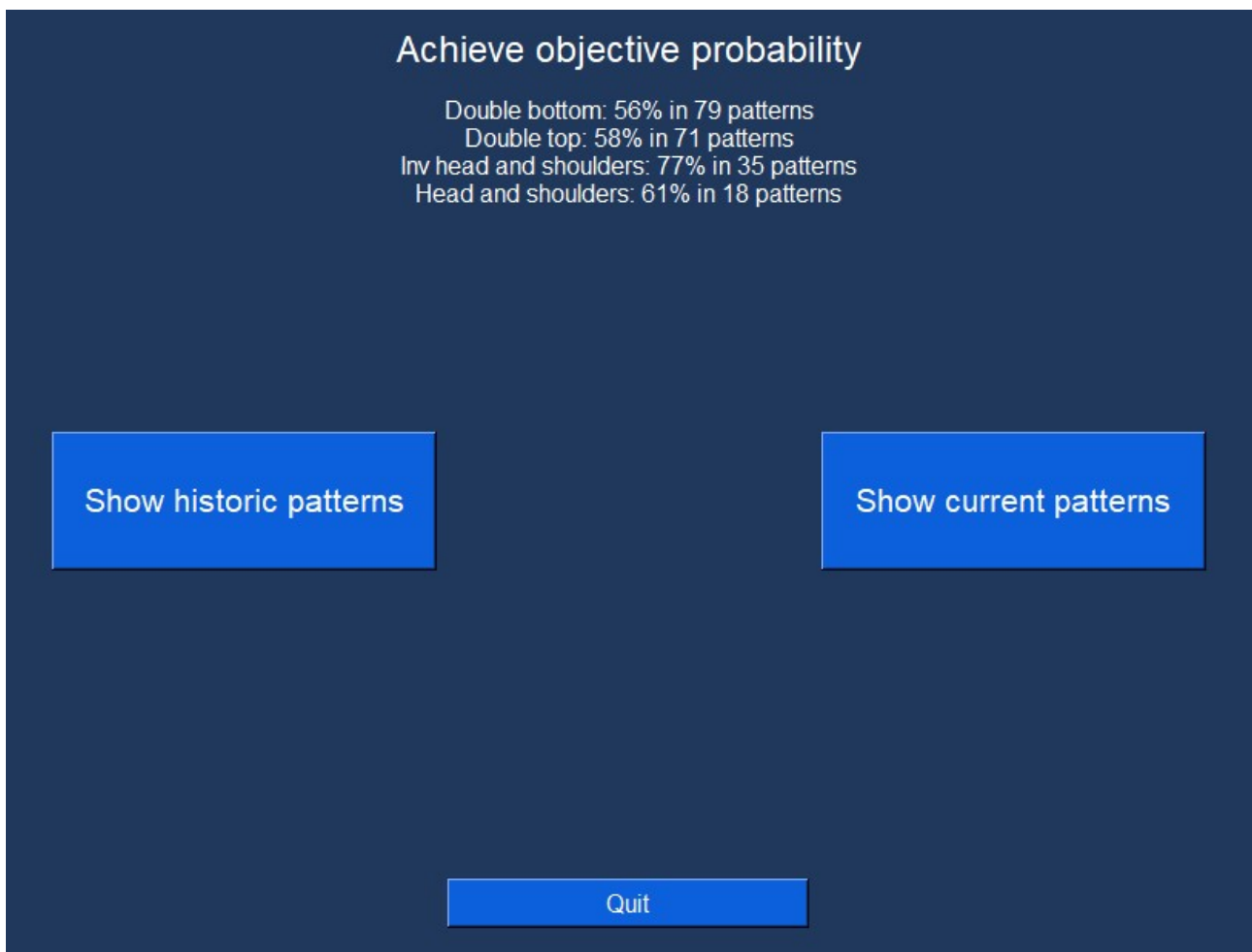


Figura A.1: Datos financieros

Achieve objective probability

Descending triangle: 25% in 4 patterns
Ascending triangle: 36% in 11 patterns

Show historic patterns

Show current patterns

Quit

Figura A.2: Datos financieros

Achieve objective probability

Double top: 67% in 83 patterns
Inv head and shoulders: 72% in 36 patterns
Head and shoulders: 58% in 24 patterns
Double bottom: 69% in 81 patterns

Show historic patterns

Show current patterns

Quit

Figura A.3: Datos tecnológicos

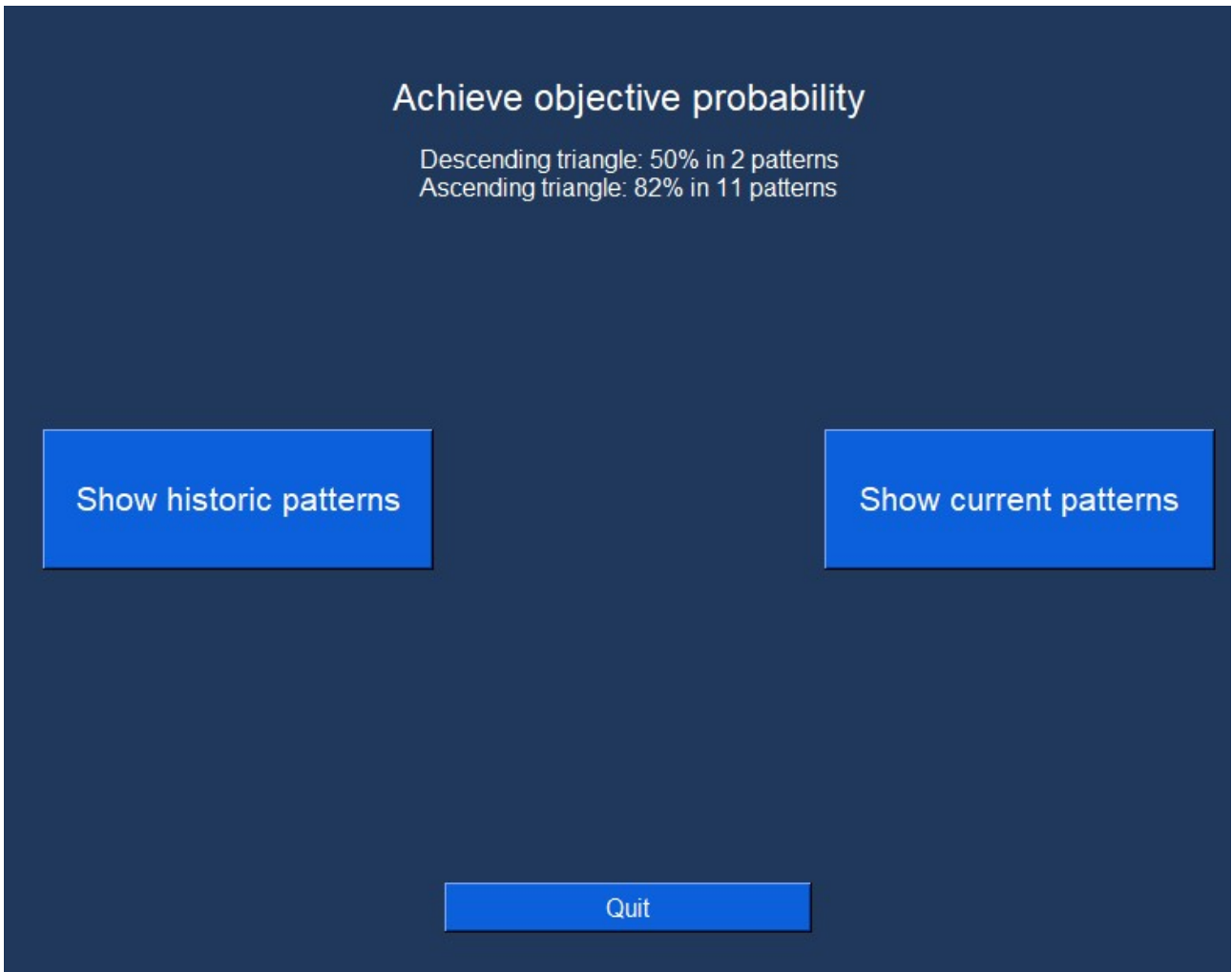
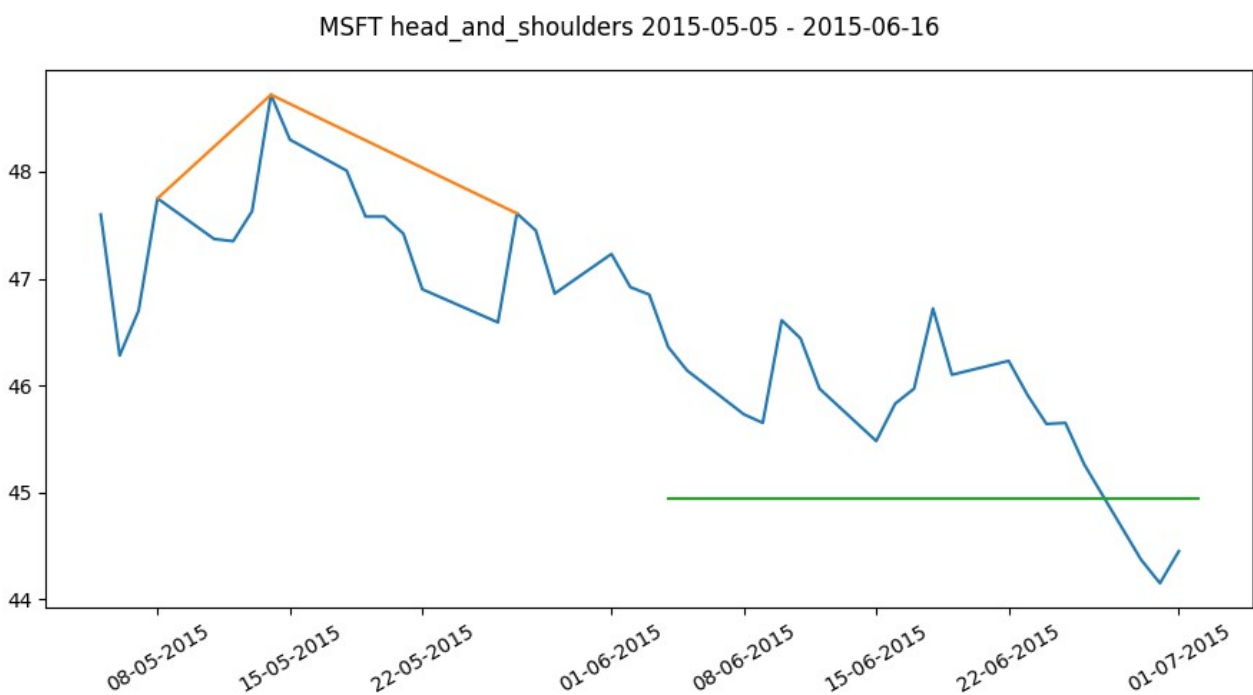
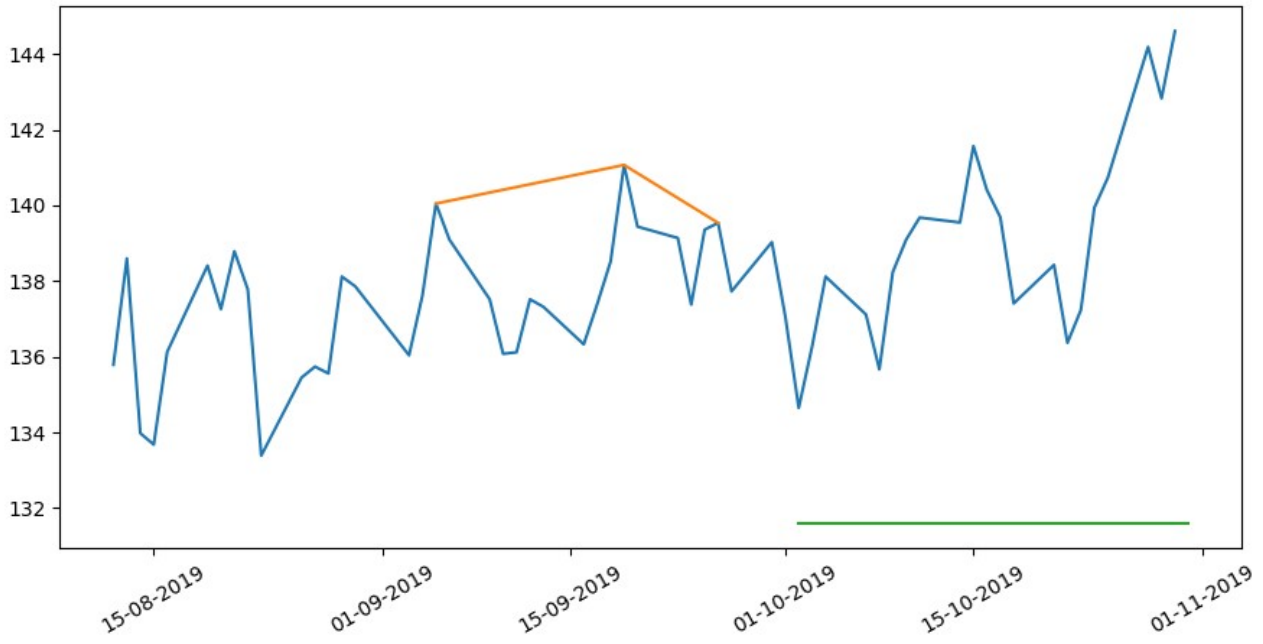


Figura A.4: Datos tecnológicos

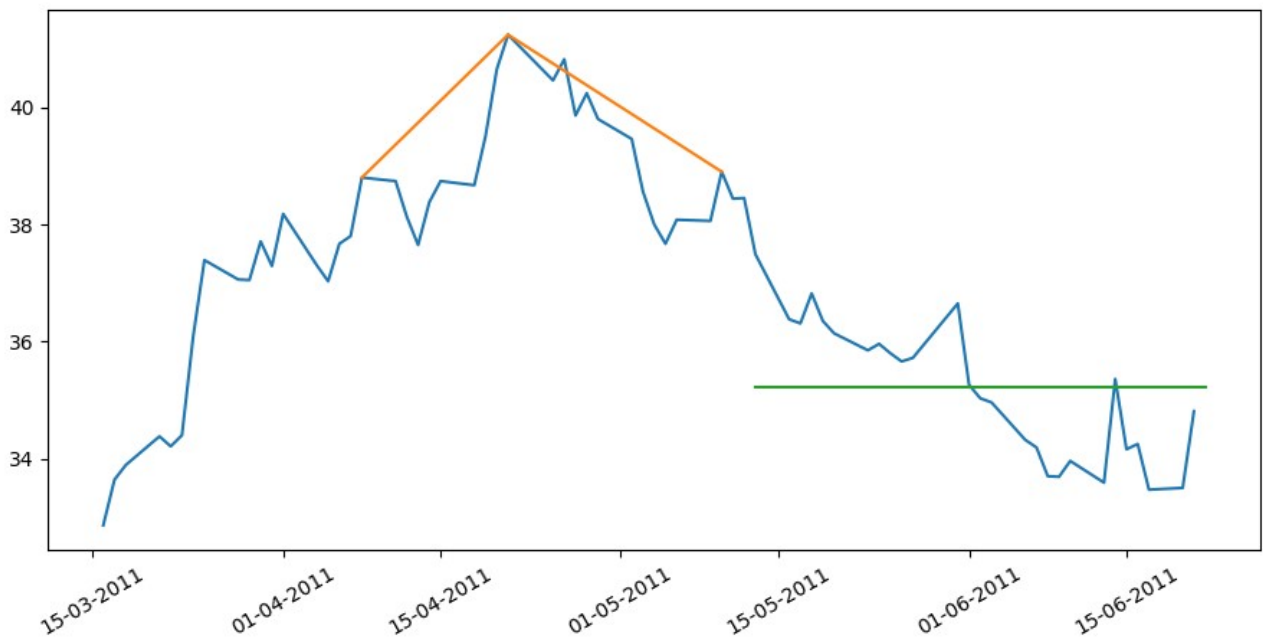
A.2.1. Resultados hombro cabeza hombro sector tecnológico



MSFT head_and_shoulders 2019-08-12 - 2019-10-07



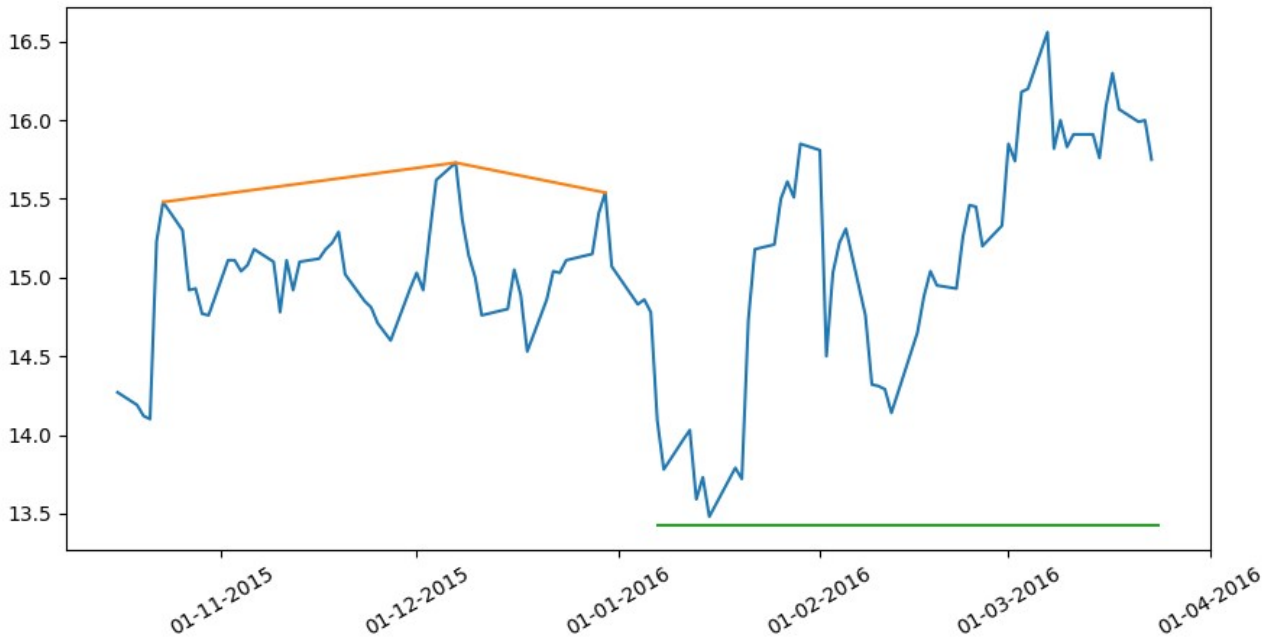
WDC head_and_shoulders 2011-03-16 - 2011-06-09



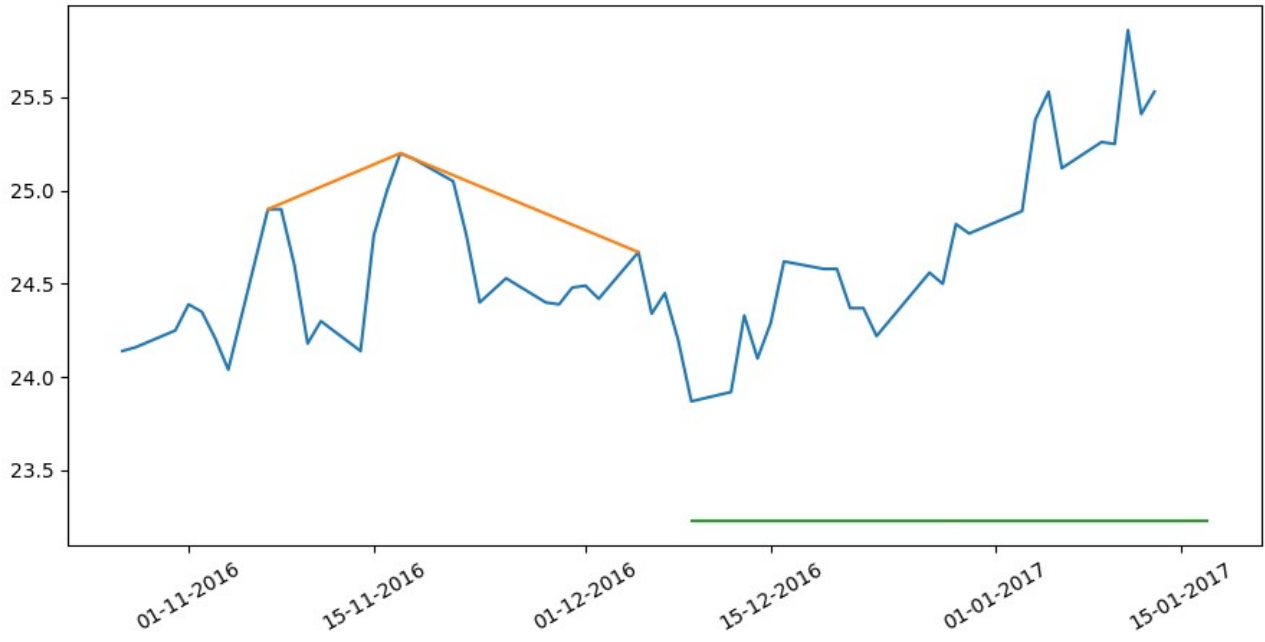
WDC head_and_shoulders 2021-11-29 - 2022-02-23



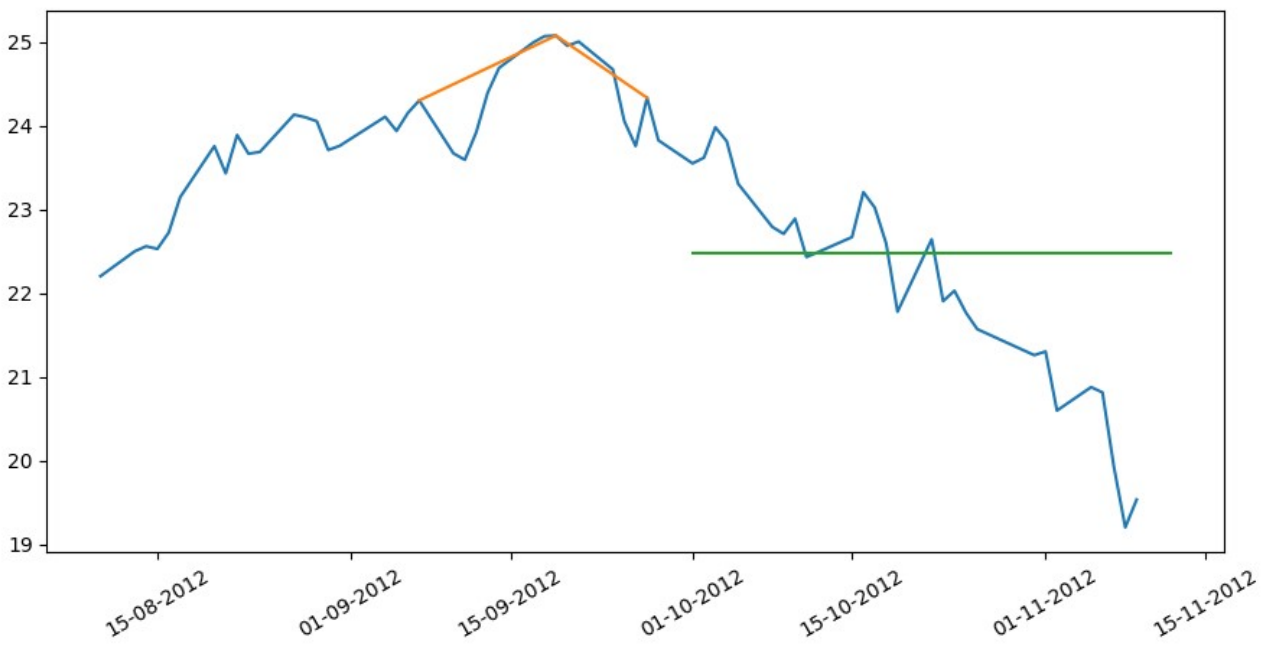
LOGI head_and_shoulders 2015-10-16 - 2016-01-12



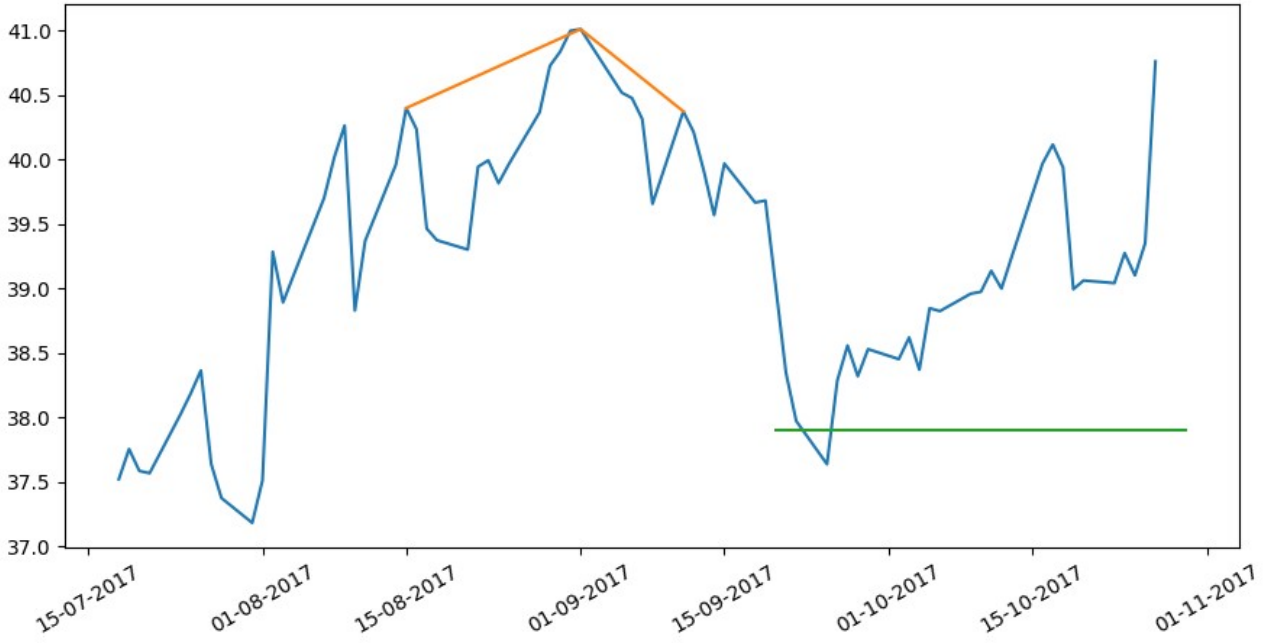
LOGI head_and_shoulders 2016-10-27 - 2016-12-22



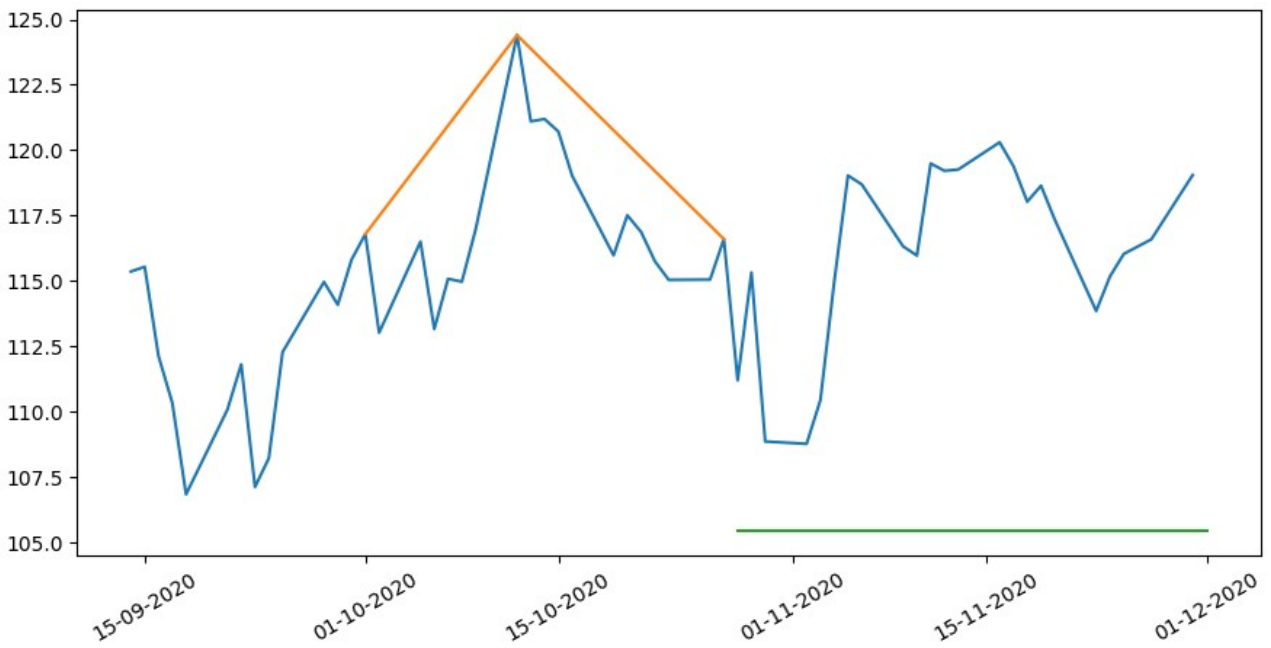
AAPL head_and_shoulders 2012-08-10 - 2012-11-06



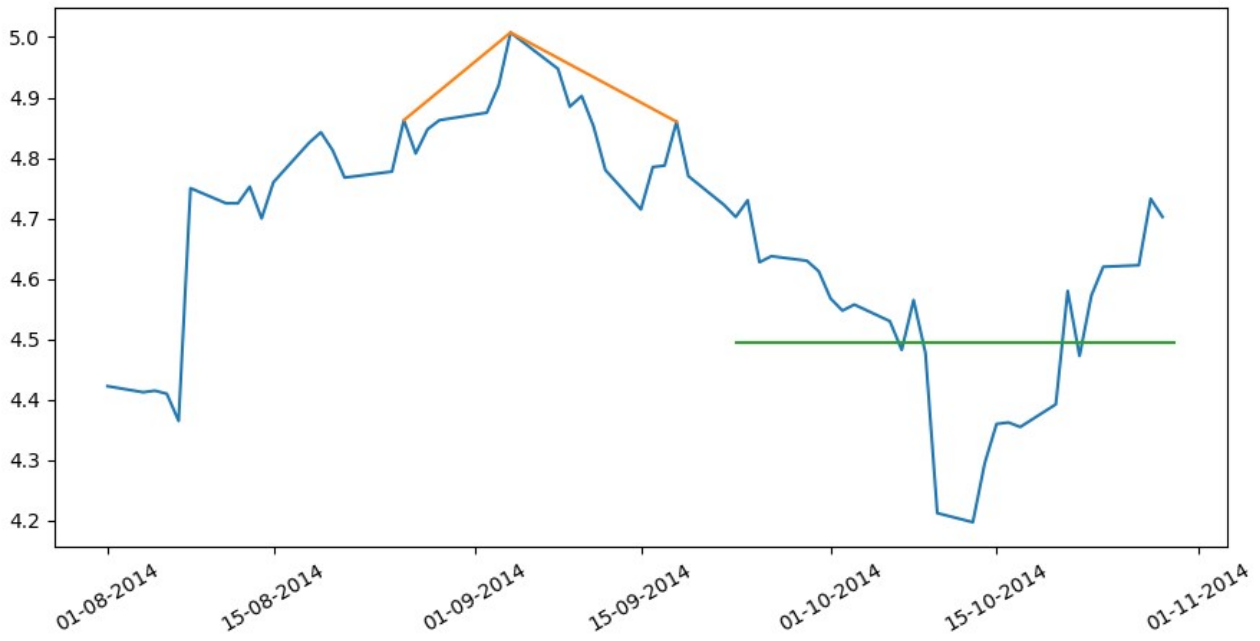
AAPL head_and_shoulders 2017-07-18 - 2017-10-10



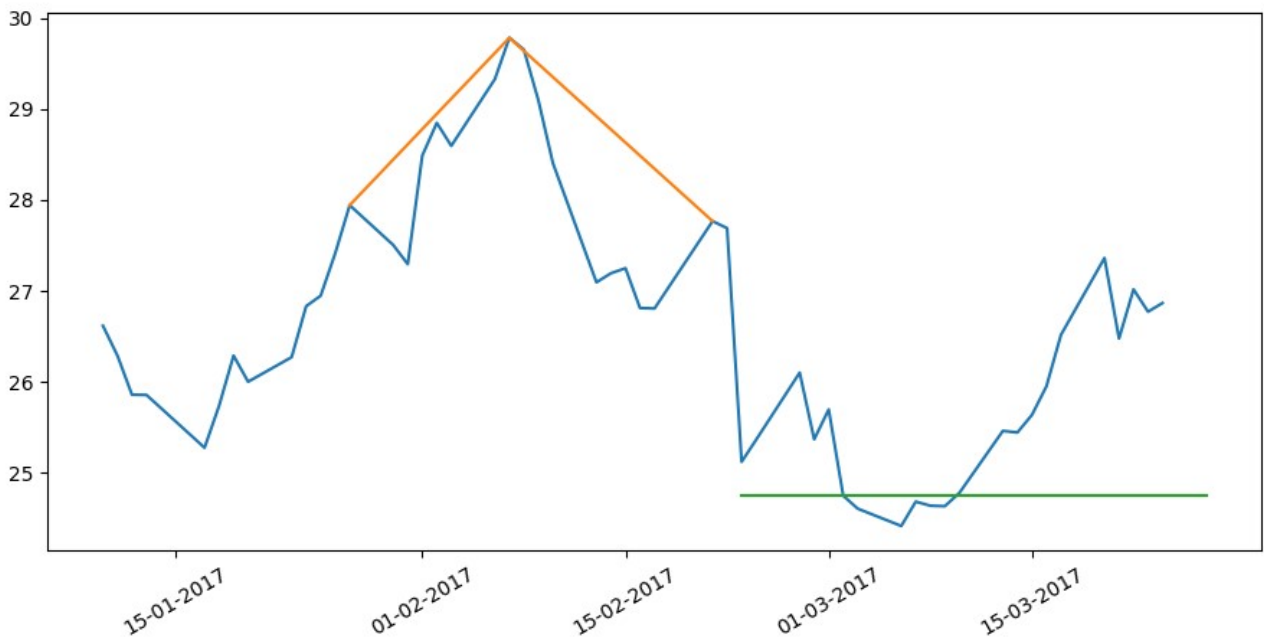
AAPL head_and_shoulders 2020-09-14 - 2020-11-06



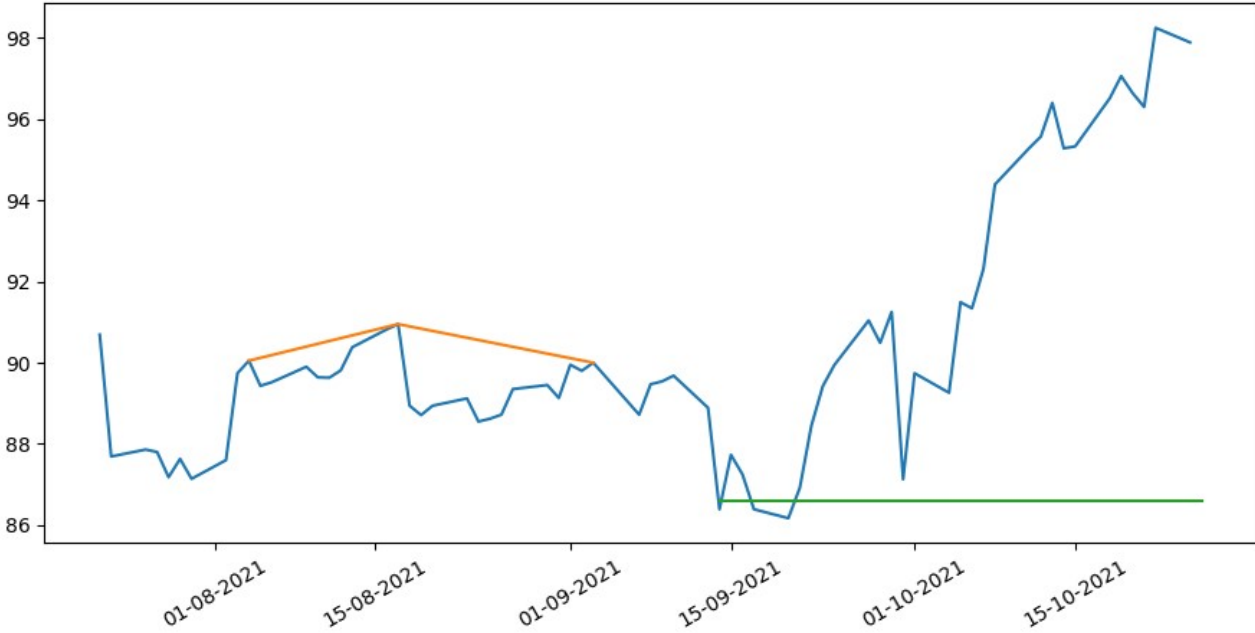
NVDA head_and_shoulders 2014-08-01 - 2014-10-24



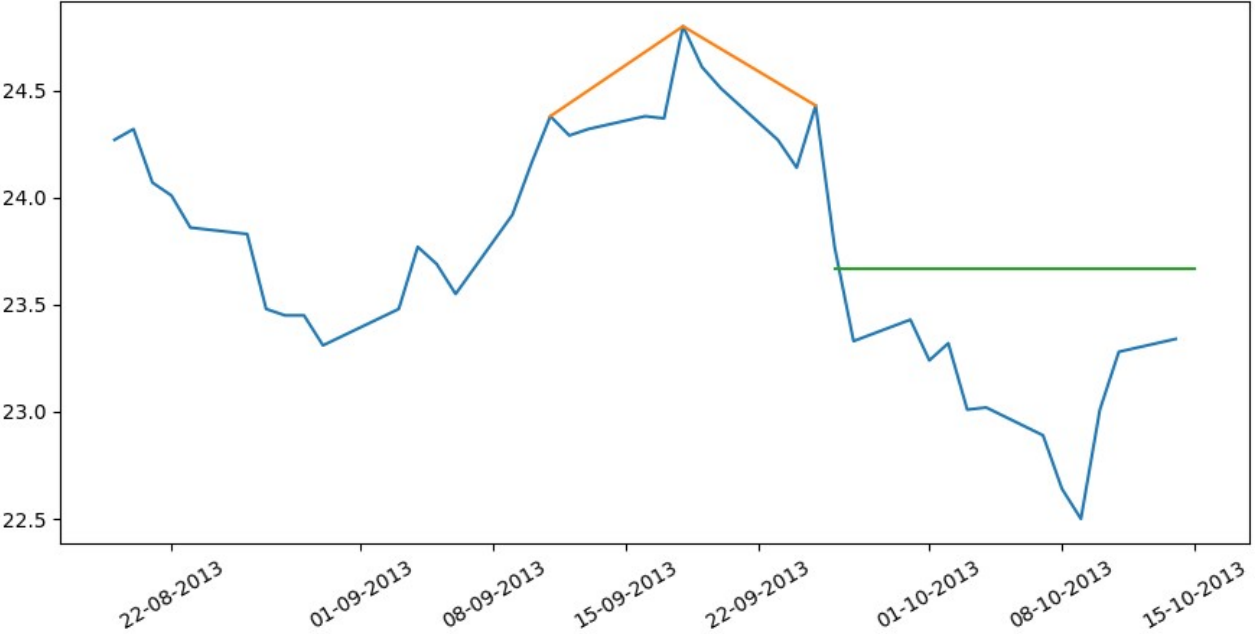
NVDA head_and_shoulders 2017-01-10 - 2017-03-08



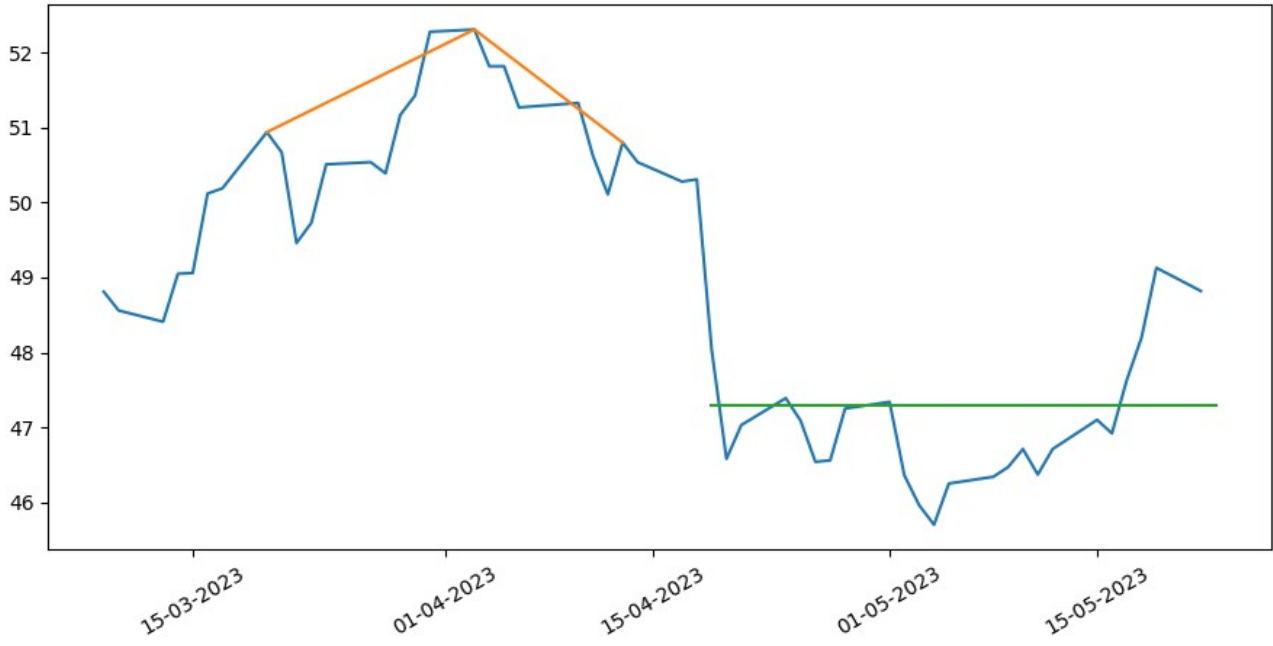
ORCL head_and_shoulders 2021-07-22 - 2021-09-16



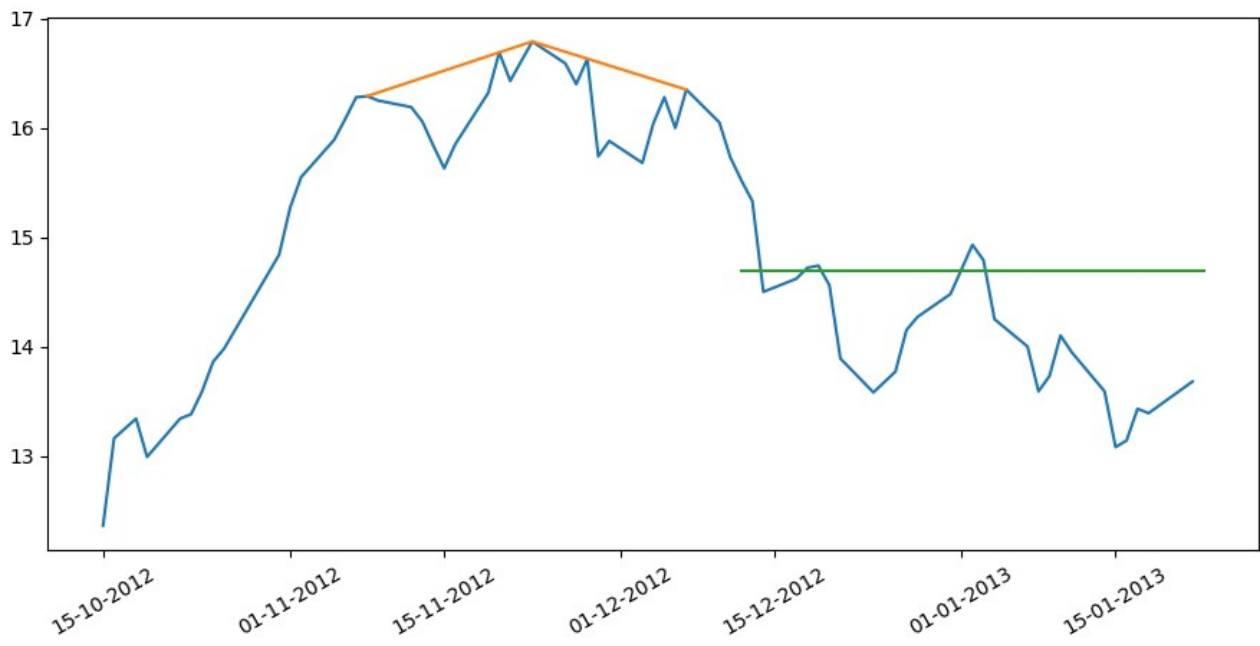
CSCO head_and_shoulders 2013-08-19 - 2013-09-30



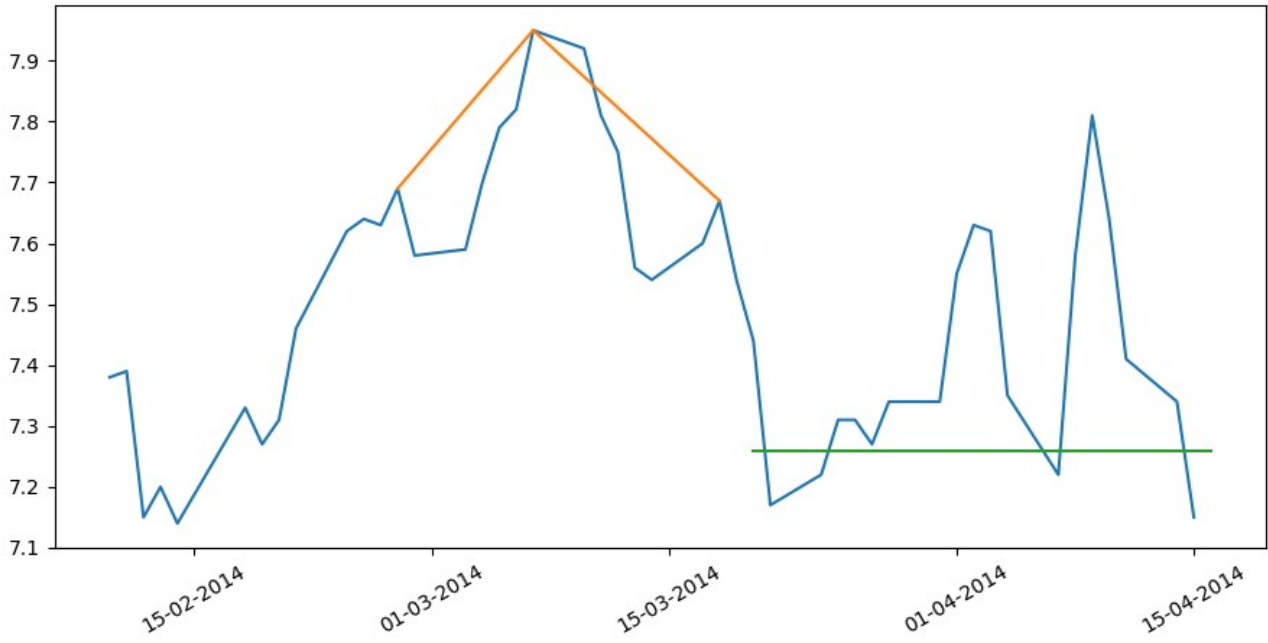
CSCO head_and_shoulders 2023-03-09 - 2023-05-04



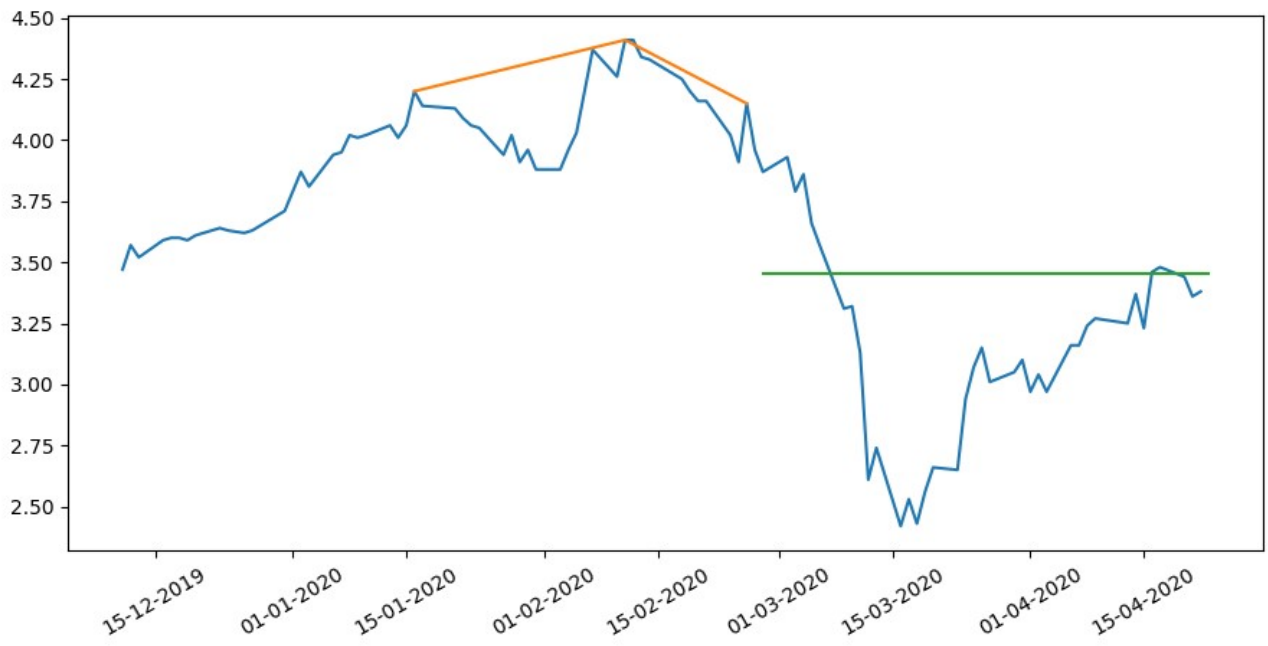
LPL head_and_shoulders 2012-10-15 - 2013-01-11



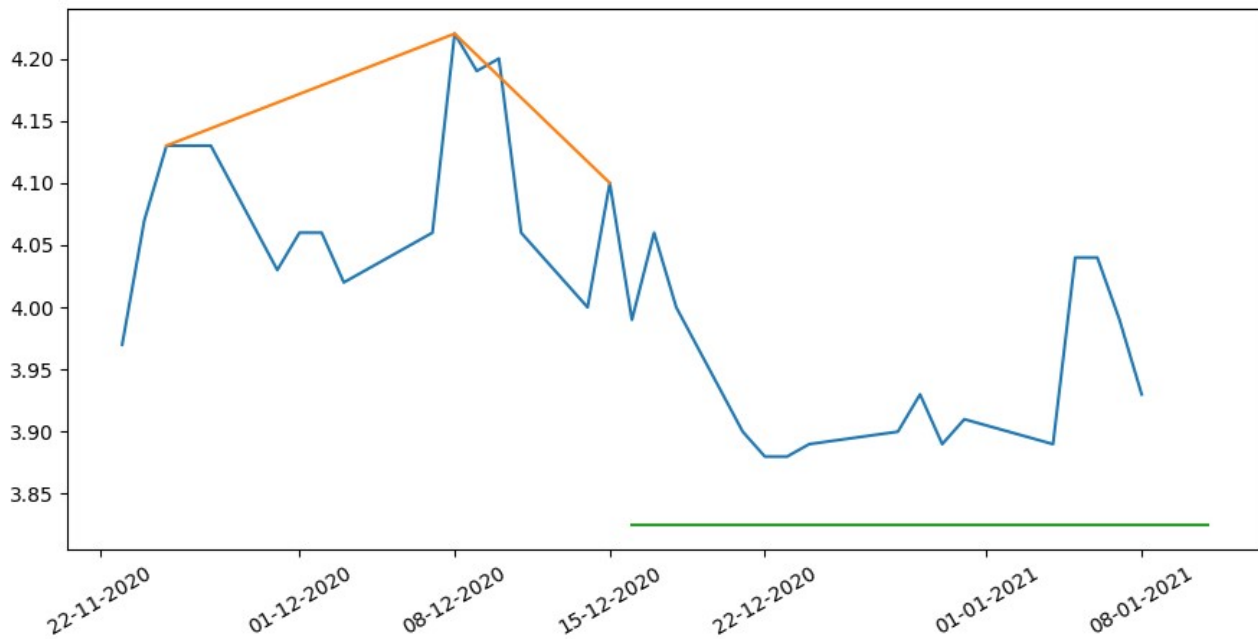
NOK head_and_shoulders 2014-02-10 - 2014-03-24



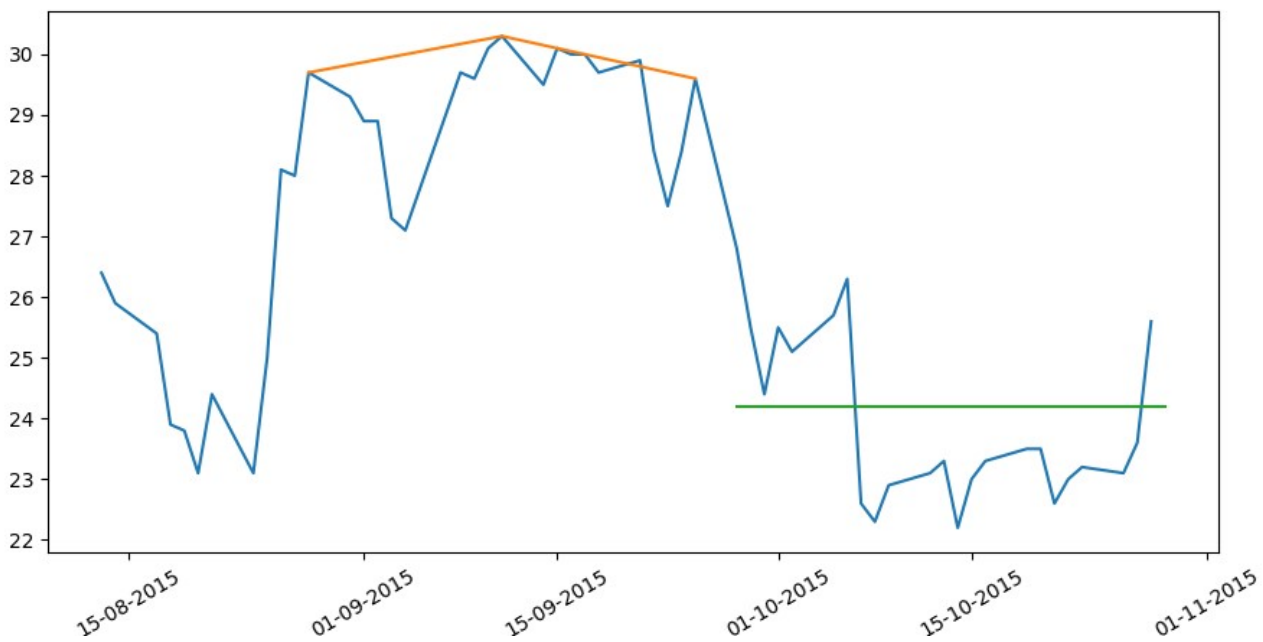
NOK head_and_shoulders 2019-12-11 - 2020-03-09



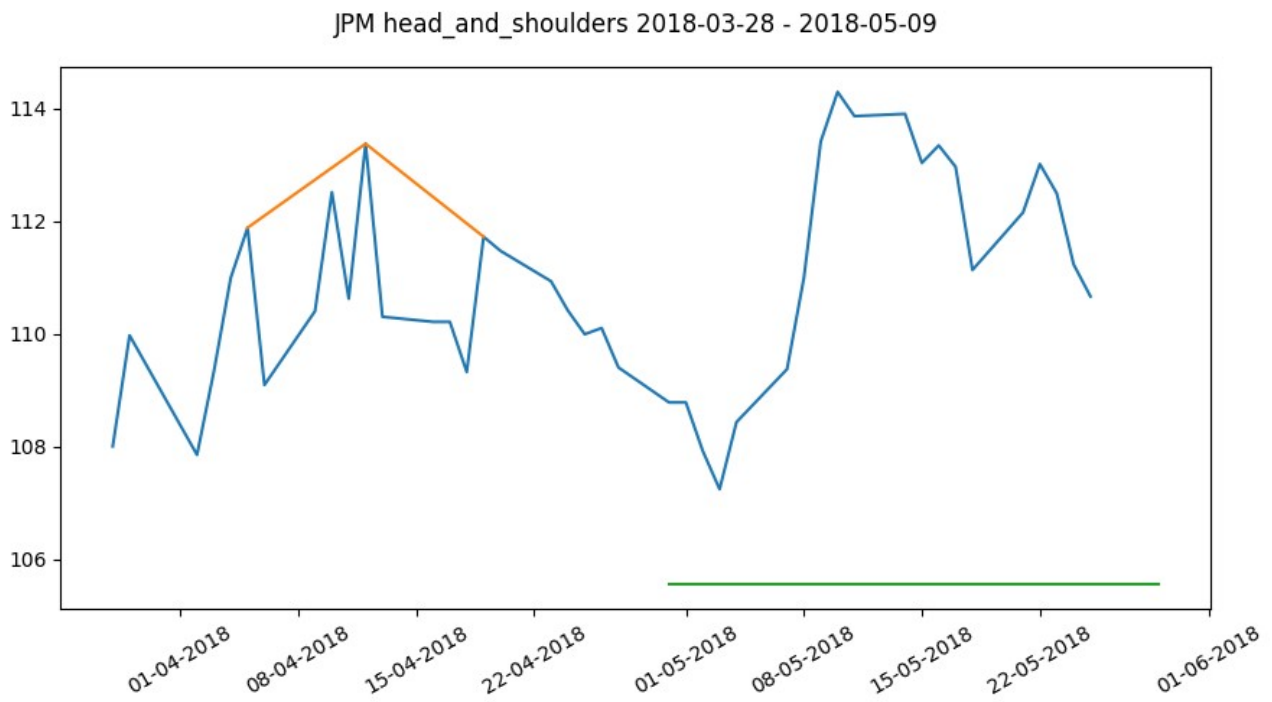
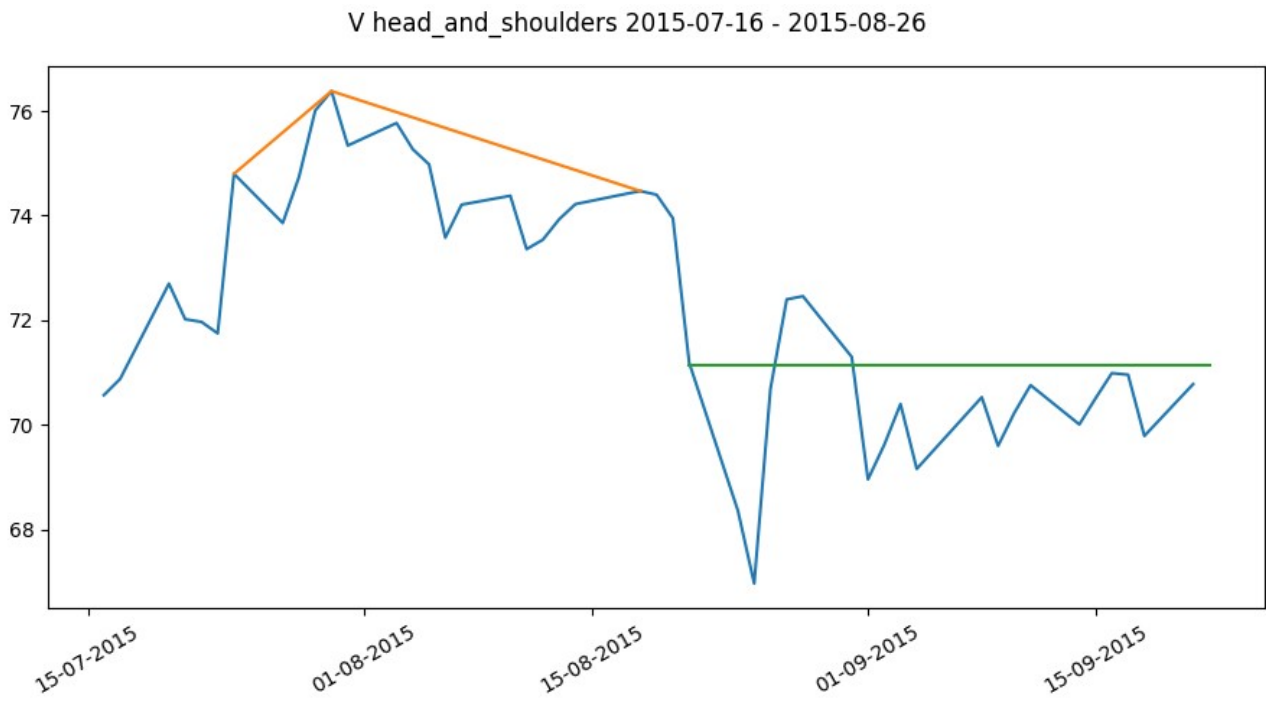
NOK head_and_shoulders 2020-11-23 - 2021-01-06



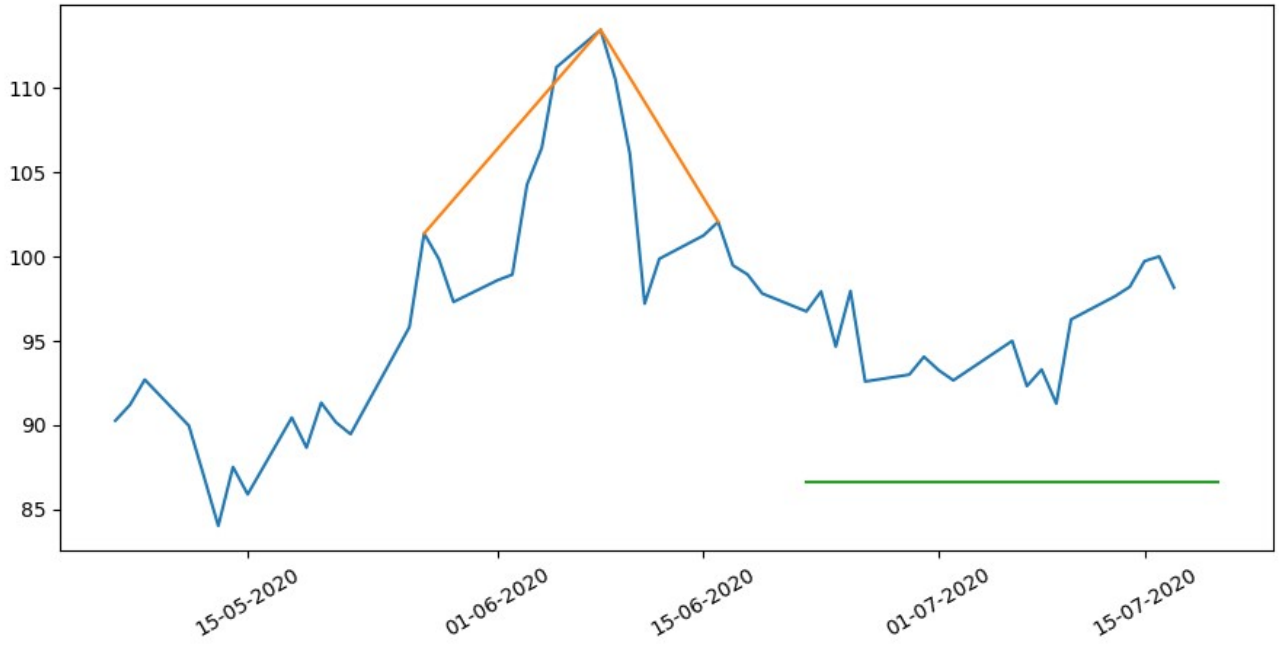
NEON head_and_shoulders 2015-08-13 - 2015-10-08



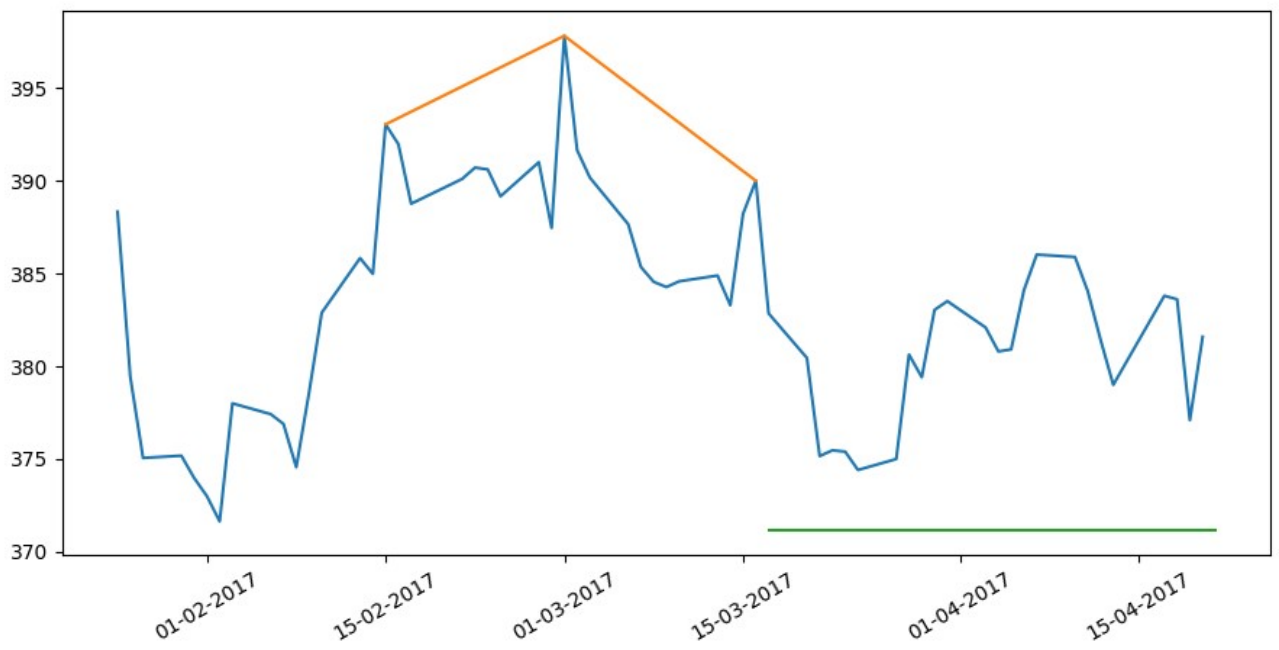
A.2.2. Resultados hombro cabeza hombro sector financiero



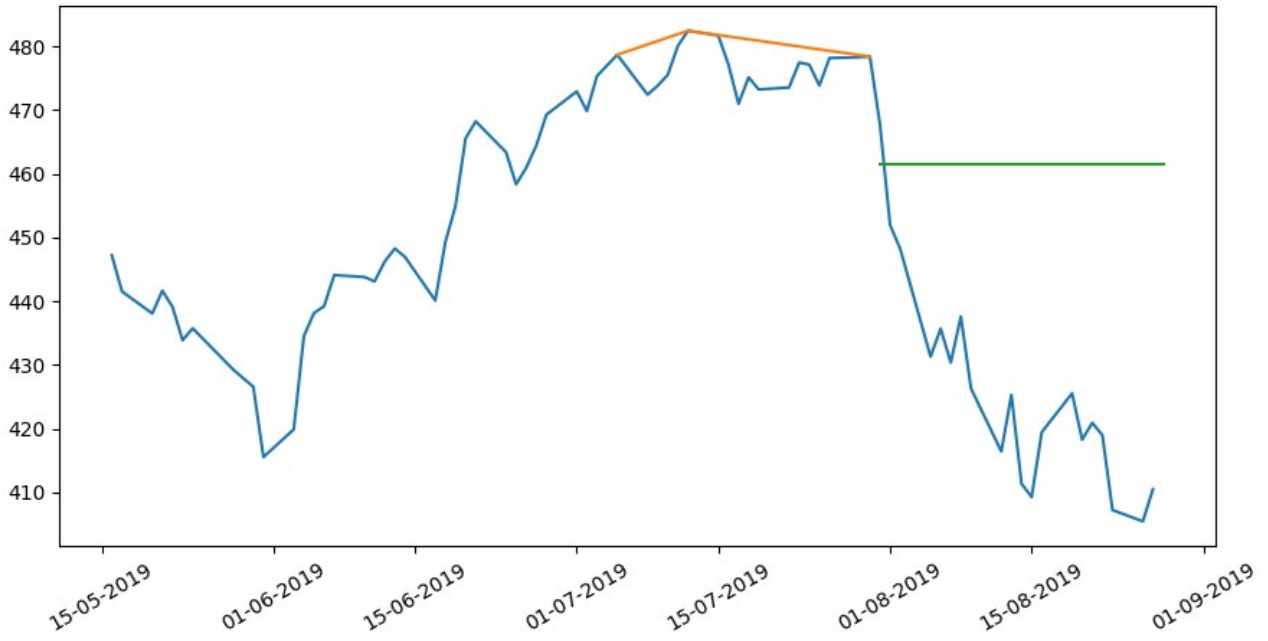
JPM head_and_shoulders 2020-05-06 - 2020-07-30



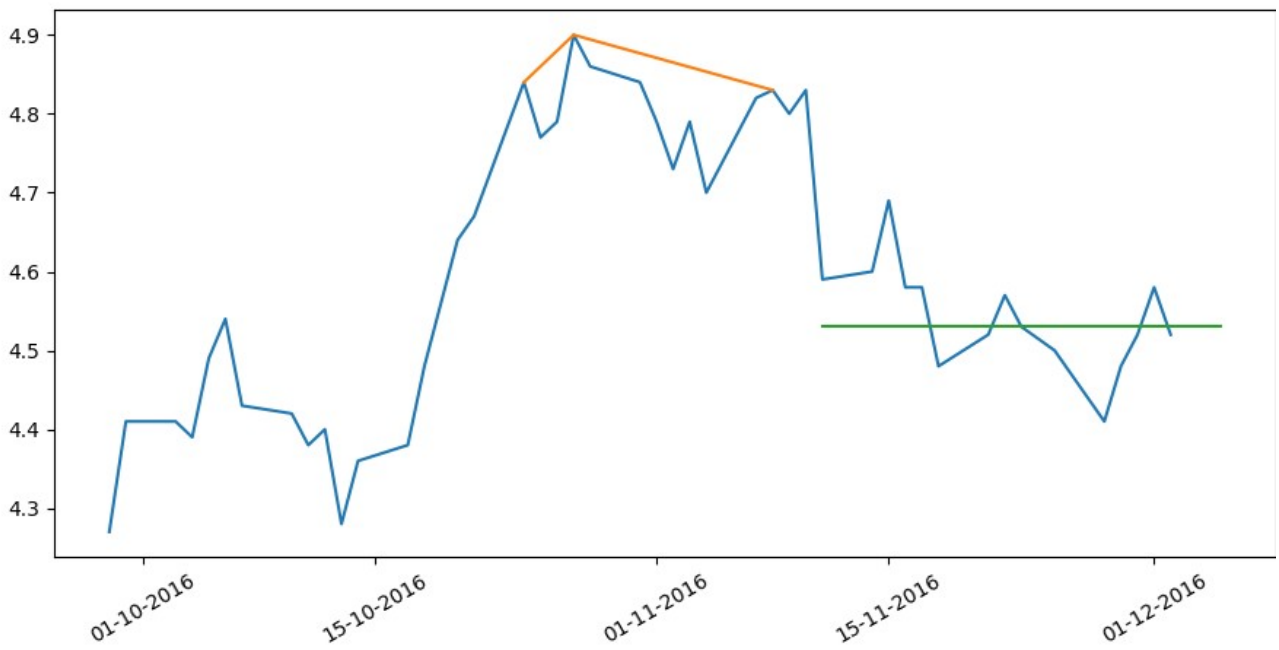
BLK head_and_shoulders 2017-01-25 - 2017-03-22



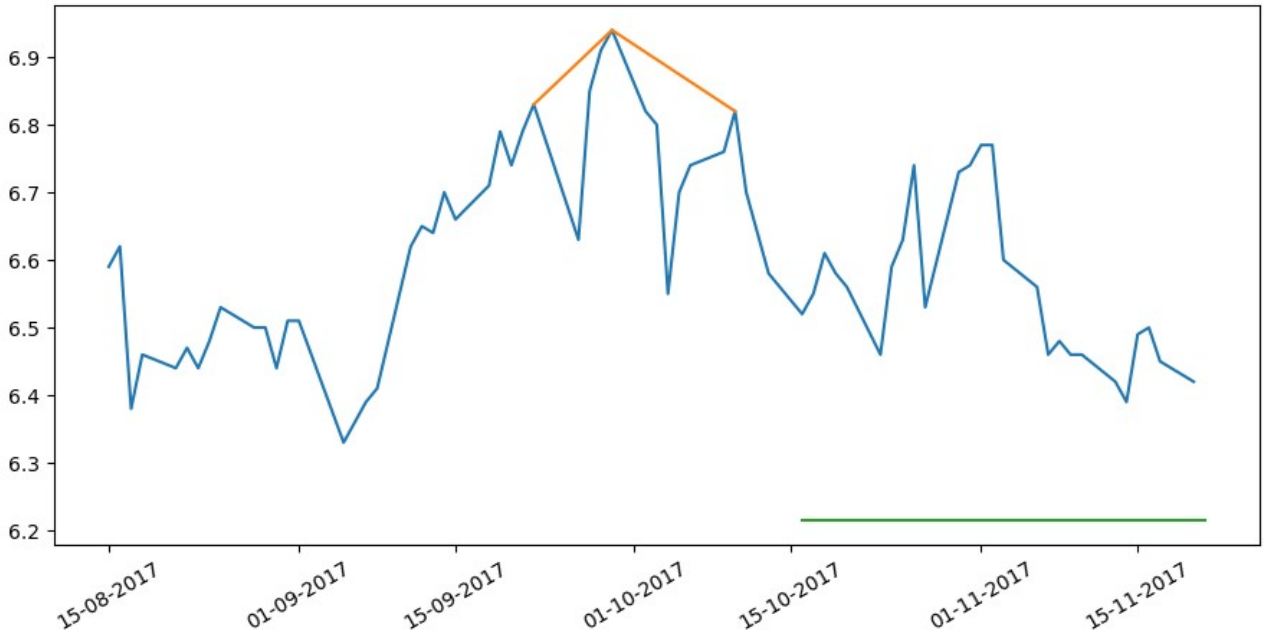
BLK head_and_shoulders 2019-05-16 - 2019-08-09



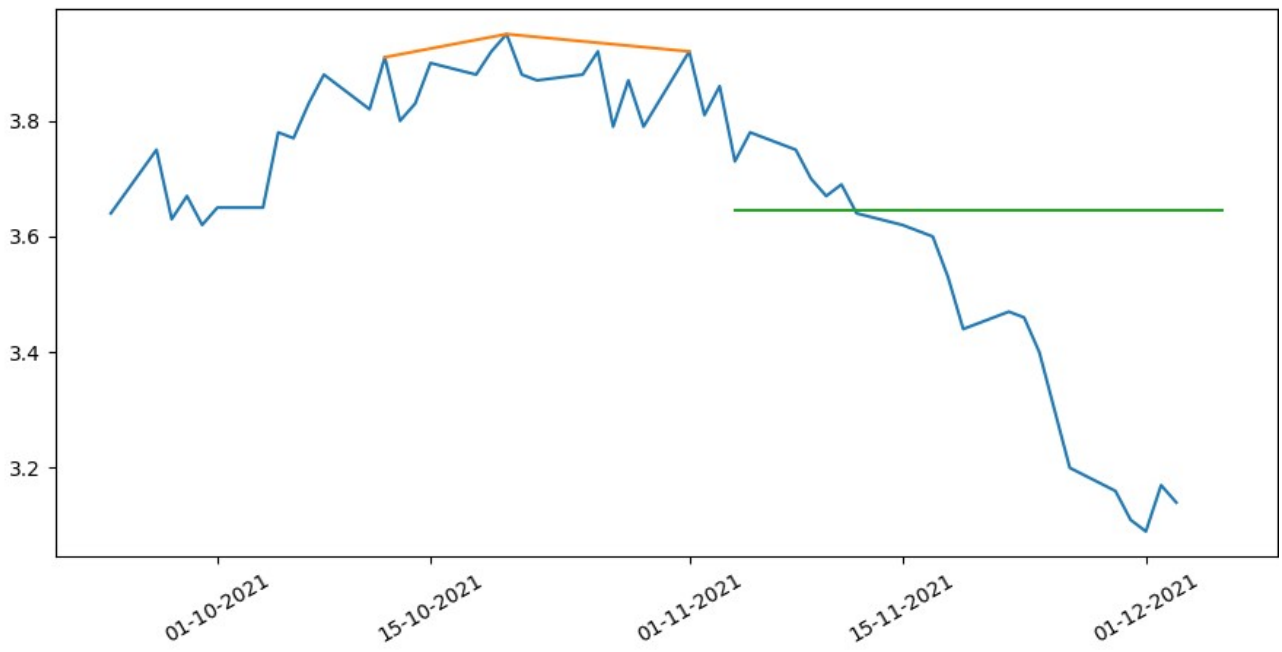
SAN head_and_shoulders 2016-09-29 - 2016-11-23



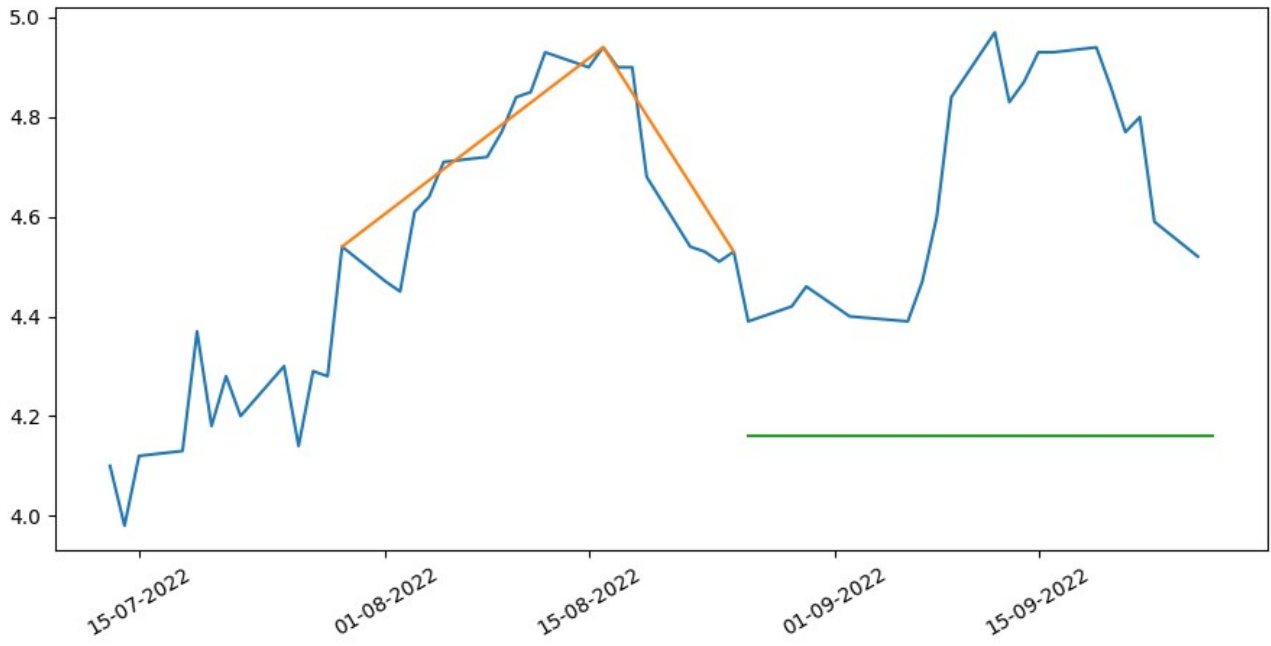
SAN head_and_shoulders 2017-08-15 - 2017-11-07



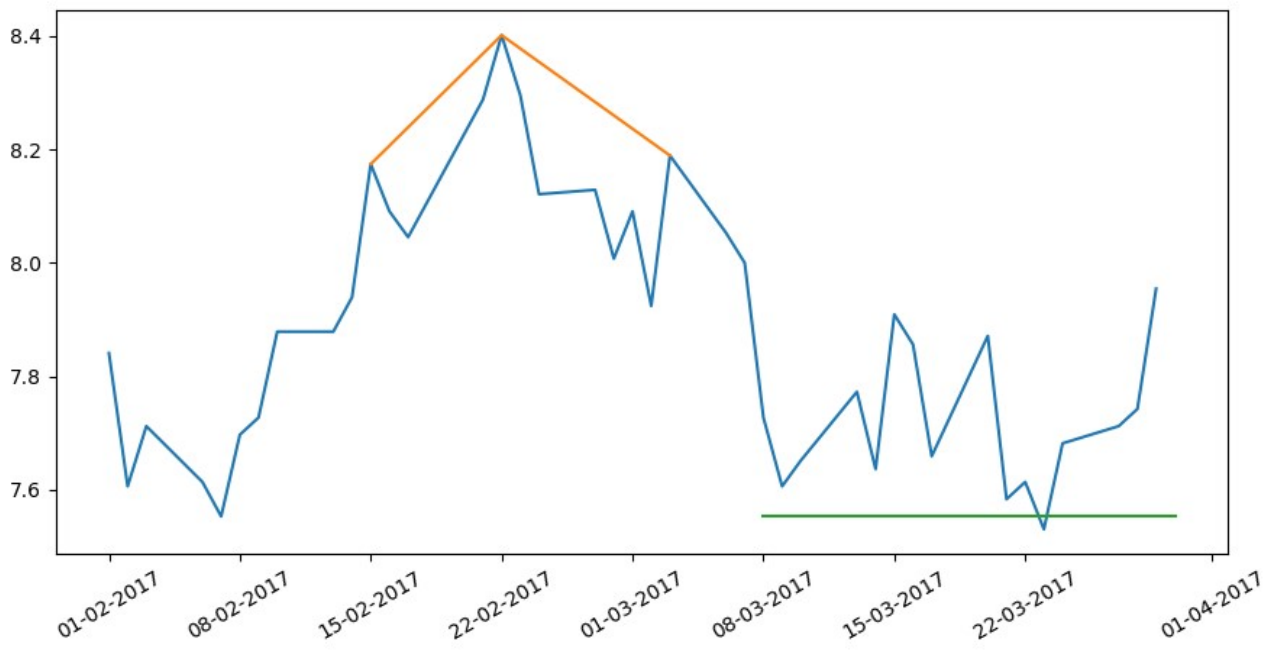
SAN head_and_shoulders 2021-09-24 - 2021-11-18



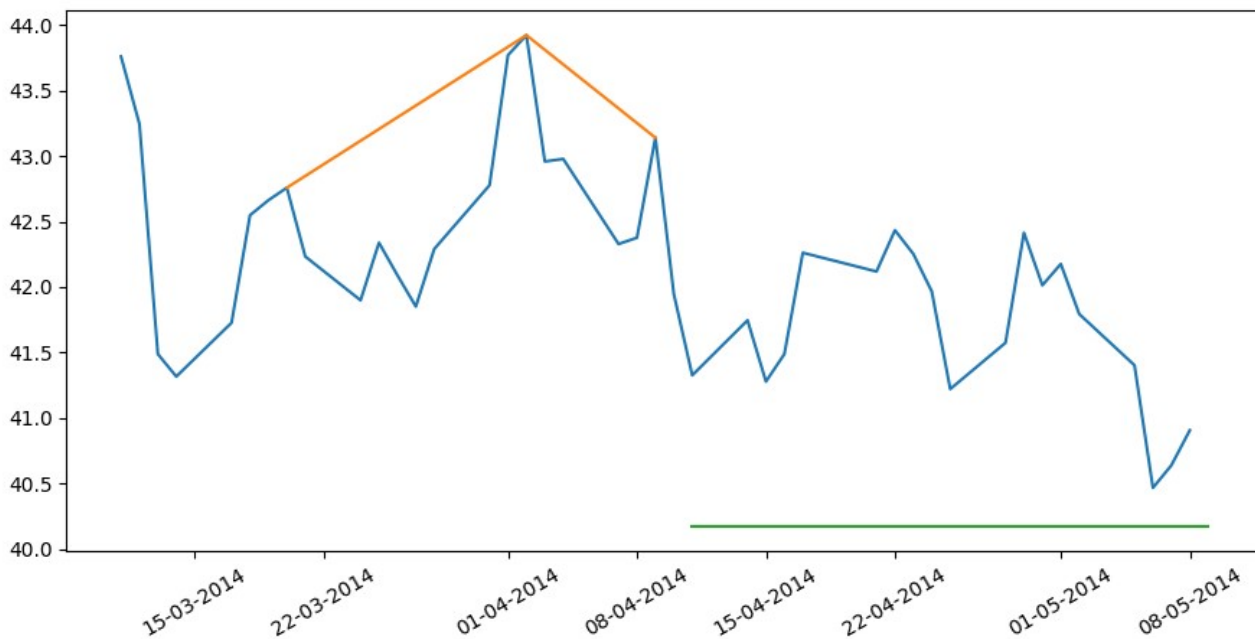
BBVA head_and_shoulders 2022-07-13 - 2022-09-07



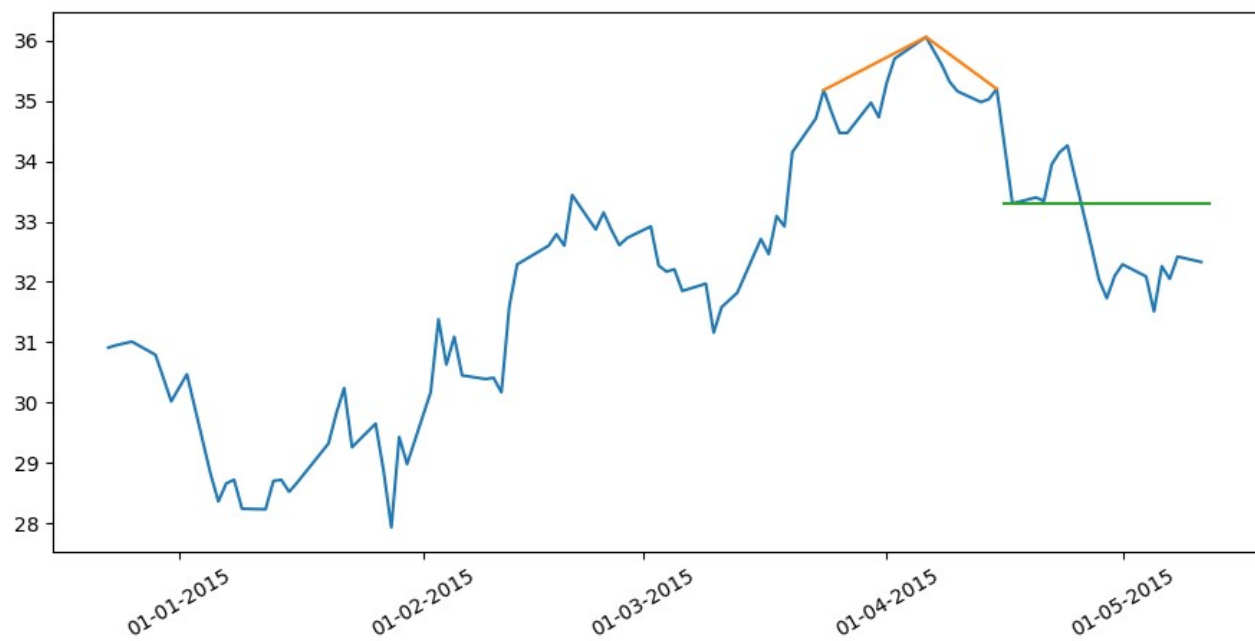
BBD head_and_shoulders 2017-02-01 - 2017-03-29



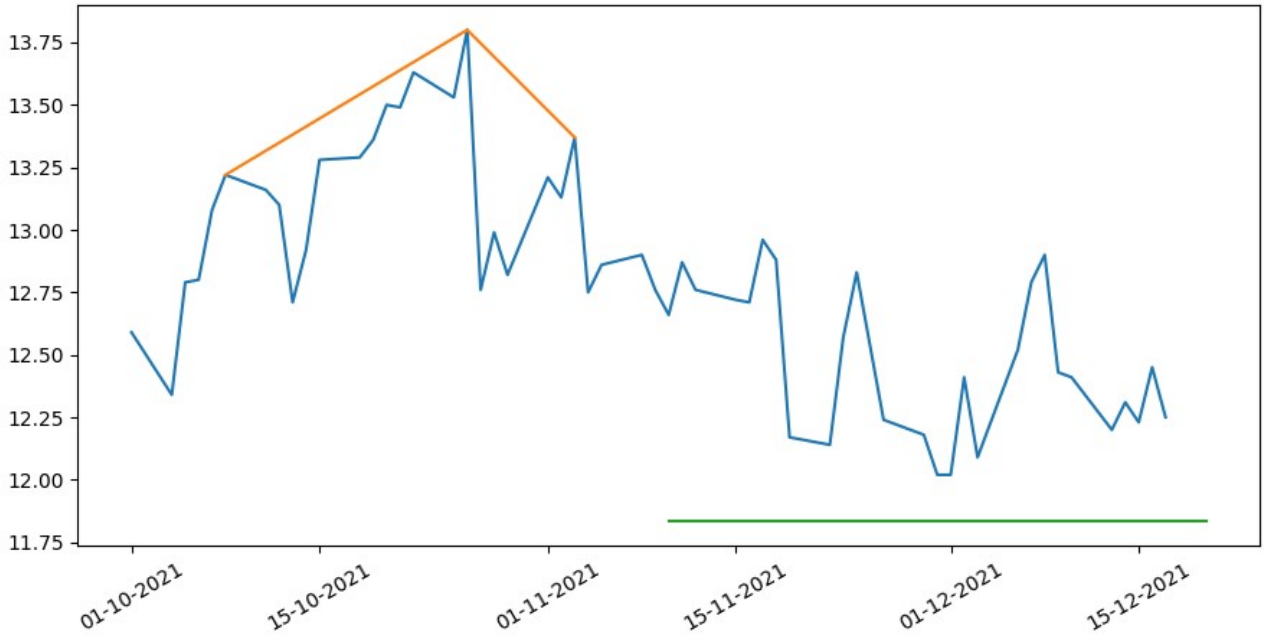
DB head_and_shoulders 2014-03-11 - 2014-05-06



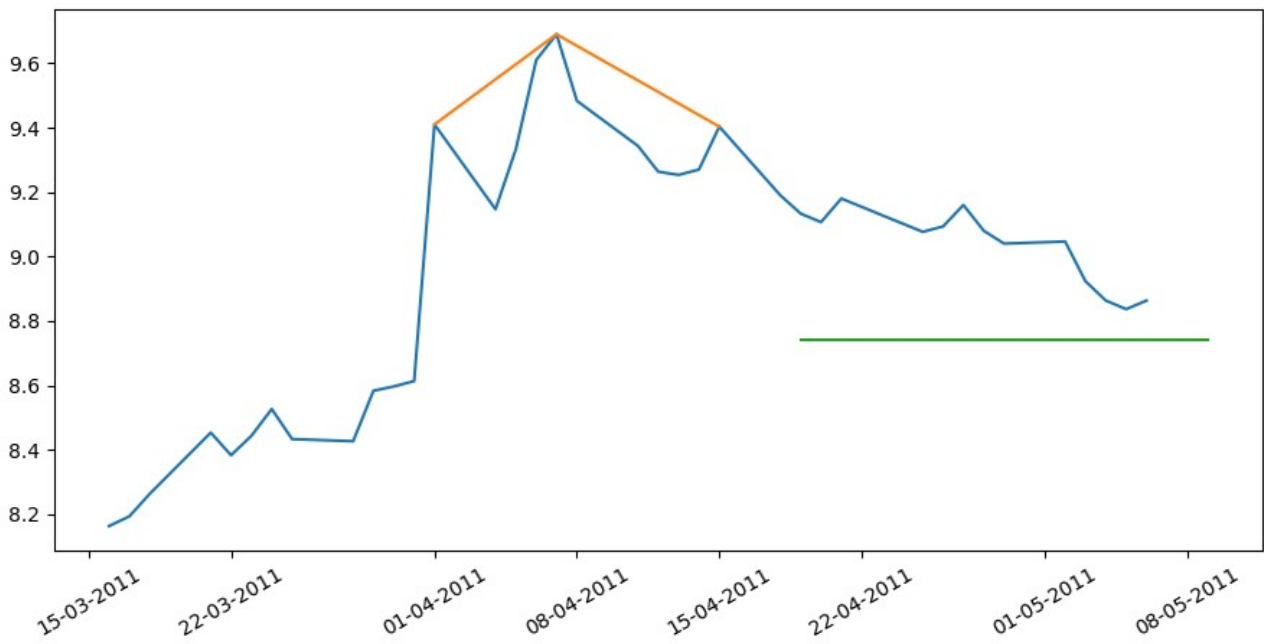
DB head_and_shoulders 2014-12-23 - 2015-06-16



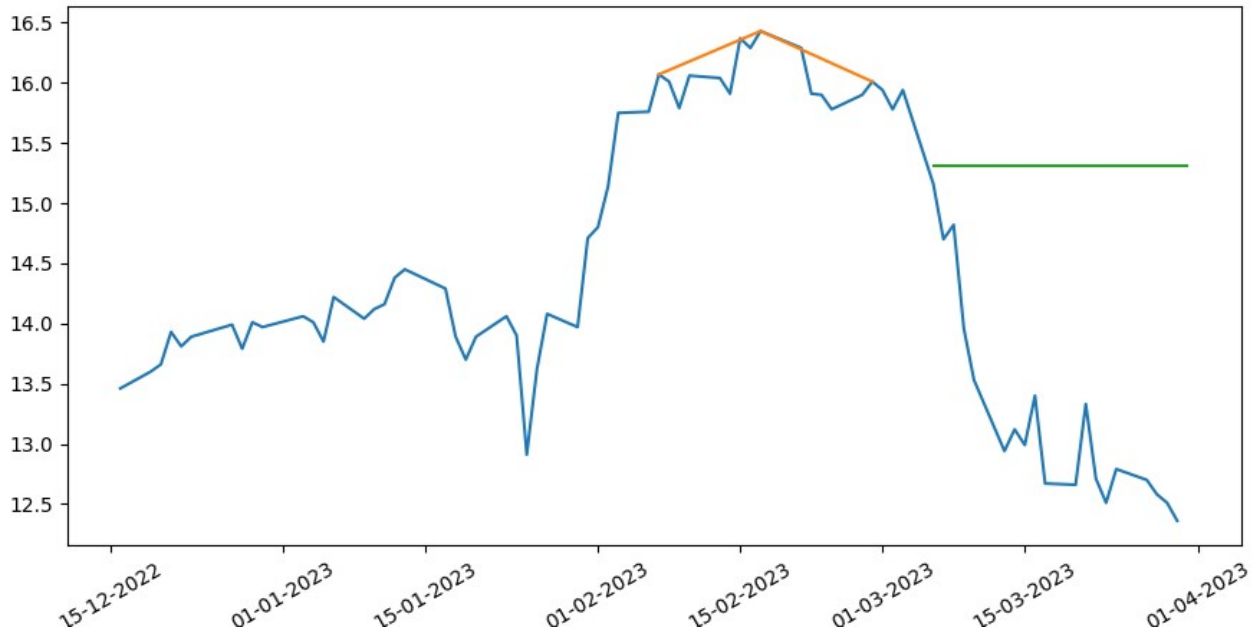
DB head_and_shoulders 2021-10-01 - 2021-11-26



NDAQ head_and_shoulders 2011-03-16 - 2011-06-09



FCF head_and_shoulders 2022-12-16 - 2023-03-15



Bibliografía

- [1] Gabriel García, *Herramienta de inteligencia artificial para detectar patrones en bolsa*, Repositorio institucional de la Universidad de La Laguna, 2022, <https://riull.ull.es/xmlui/handle/915/29420>
- [2] Achelis, S.B. (1995). *Technical analysis from A to Z*. (Manuel Cabedo, Trad.). Mc Graw-Hill
- [3] Torre Gallegos, A.d.l. y Ruiz Martínez, R.J. (1999). *El papel del análisis técnico en la filosofía del inversor medio*. Boletín de estudios económicos, 54 (167), 361-387.
- [4] Ni, J., Zhang, C. (2005). *An Efficient Implementation of the Backtesting of Trading Strategies*. In: Pan, Y., Chen, D., Guo, M., Cao, J., Dongarra, J. (eds) *Parallel and Distributed Processing and Applications*. ISPA 2005. Lecture Notes in Computer Science, vol 3758. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11576235_17
- [5] Farias Nazário, Rodolfo Toríbio & e Silva, Jéssica Lima & Sobreiro, Vinicius Amorim & Kimura, Herbert, 2017. *A literature review of technical analysis on stock markets*, "The Quarterly Review of Economics and Finance, Elsevier, vol. 66(C), pages 115-126.
- [6] Kim, S., Lee, H., Ko, H., Jeong, S., Byun, H., & Oh, K. (2018). *Pattern Matching Trading System Based on the Dynamic Time Warping Algorithm*. Sustainability, 10(12), 4641. <https://doi.org/10.3390/su10124641>
- [7] Fu-Lai Chung, Tak-Chung Fu, V. Ng and R. W. P. Luk, "An evolutionary approach to pattern-based time series segmentation," in IEEE Transactions on Evolutionary Computation, vol. 8, no. 5, pp. 471-489, Oct. 2004, DOI: 10.1109/TEVC.2004.832863.
- [8] Dong, M., & Zhou, X. (2002). *EXPLORING THE FUZZY NATURE OF TECHNICAL PATTERNS OF U . S STOCK MARKET*.
- [9] Y. -L. Hsu, C. -L. Chu, Y. -J. Tsai and J. -S. Wang, *An Inertial Pen With Dynamic Time Warping Recognizer for Handwriting and Gesture Recognition*," in IEEE Sensors Journal, vol. 15, no. 1, pp. 154-163, Jan. 2015, doi: 10.1109/JSEN.2014.2339843.
- [10] Moisés Díaz, Gioele Crispo, Antonio Parziale, Angelo Marcelli, Miguel A. Ferrer: *Writing Order Recovery in Complex and Long Static Handwriting*. Int. J. Interact. Multim. Artif. Intell. 7(4): 171 (2022)
- [11] A. Sharabiani, H. Darabi, A. Rezaei, S. Harford, H. Johnson and F. Karim, *Efficient Classification of Long Time Series by 3-D Dynamic Time Warping*," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 10, pp. 2688-2703, Oct. 2017, doi: 10.1109/TSMC.2017.2699333.

[12] Repositorio GitHub: <https://github.com/AitorAlonsoMelian/PerseumAI>

[13] Aplicación PerseumAI: <https://drive.google.com/file/d/1HttBz1FRj8d0pMoaUGFGgnolzLUdVWx-/view?usp=sharing>