Universitat Oberta de Catalunya

Trabajo de fin de máster del Máster en Ciencia de Datos

Optimización de cartera de activos financieros aplicando aprendizaje automático

---

Autor:
Caparrini López Antonio

Director:
Dr. Escayola Mansilla Jordi

3 de enero de 2021

Resumen

El valor temporal del dinero siempre ha empujado a los poseedores de capital a invertir para conseguir rentabilidad. Estas inversiones generalmente buscan maximizar la rentabilidad minimizando la cantidad de riesgo, lo cuál ha sido estudiado ampliamente.

Los mercados de capital han crecido enormemente durante el último siglo y la información que se tiene de las empresas y el mercado además de tener un gran volumen, no deja de crecer. Para poder reducir está ingente cantidad de datos a información que pueda usarse para tomar decisiones numerosos estudios han identificado lo que denominan *factores*. Un factor busca identificar una característica común entre activos de manera que permita identificar cuales producen mayor rentabilidad.

Actualmente, un estilo de inversión cada vez más frecuente son los fondos que replican un índice (gestión pasiva), generando una exposición a un mercado en concreto (Ej: *SP500*) que históricamente en su conjunto siempre ha producido rentabilidad, reduciendo el riesgo mediante la diversificación, al estar el índice compuesto por numerosos activos. Además este estilo de gestión tiene unos costes bajos que la hacen atractiva para los inversores.

Por otro lado, tenemos la gestión activa, donde los fondos son gestionados de manera que mediante análisis y criterios propios buscan conseguir una rentabilidad mayor que la del mercado a cambio de mayores costes de gestión.

La evolución tecnológica reciente (tanto *hardware* como *software*) permite resolver problemas y generar modelos estadísticos de aprendizaje automático complejos que utilicen gran cantidad de datos. Mediante estas técnicas se pueden buscar patrones comunes en los factores para facilitar de forma automática el análisis de los activos óptimos para una cartera de inversión. Estos modelos se utilizan a día de hoy para automatizar el proceso de seleccionar los activos lo que se denomina *Smart Beta* y que tienen menores costes que la gestión activa y mayores rentabilidades que la inversión en índices.

El propósito de este proyecto es utilizar aprendizaje automático para realizar un modelo que seleccione, a partir de las características de los activos (reflejadas en los factores presentes en la literatura), los que tendrán mejor desempeño para añadirlos a la cartera.

Palabras clave: Gestión de activos, optimización cartera, aprendizaje automático, modelo multifactor.

Índice general

[1] *Capital Asset Pricing Model*

Índice de figuras

Listado de acrónimos

ML	*Machine Learning*
CAPM	*Capital Asset Pricing Model*
SP500	*S&P 500 Index*
SVM	*Support Vector Machine*
EBIT	*Earnings before interest and taxes*
EBITDA	*Earnings before interest, taxes, depreciation and amortization*
dt	*Decision tree*
rf	*Random forest*
xgb	*XGBoost*
AWS	*Amazon Web Services*

Capítulo 1

Introducción

La generación de carteras de activos financieros es un problema que lleva años tratándose. El auge en el volumen de datos y en la capacidad de procesamiento ha propiciado nuevas soluciones y acceso al inversor particular de las herramientas utilizadas por las grandes firmas de inversión.

En este trabajo se plantea generar una cartera de acciones pertenecientes al SP500[1] utilizando factores presentes en la literatura y técnicas de aprendizaje automático optimizando los hiperparámetros.

En esta sección se introduce conceptos básicos de la creación de carteras así como del aprendizaje automático concluyendo con la motivación y estructura de la memoria. Más detalles sobre estos puntos se pueden encontrar en la sección 2.

1.1. Contexto y justificación del Trabajo

El contexto en el que se encuentra englobado el trabajo implica varios campos del conocimiento. En primer lugar los activos financieros de renta variable, las teorías y factores utilizados para generar carteras y el aprendizaje automático.

1.1.1. Acciones - Renta Variable

La renta variable es la inversión en la que la recuperación del capital invertido y la rentabilidad obtenida no están garantizadas.

El más claro ejemplo son las acciones (*stocks*). Una acción representa la posesión de una fracción de una empresa. En este caso nos centramos en las acciones ordinarias en las que el poseedor tiene los siguientes derechos:

- Recibir dividendos.
- Participar en las votaciones.

[1] *S&P 500 Index*

- En caso de bancarrota recibir la parte proporcional a la fracción que se posee una vez se han liquidado todas las obligaciones.

Simplificando y sin tener en cuenta los dividendos, la principal forma de tener ganancias de capital con este tipo de activos es con el incremento en el valor de la acción (vendiendo la acción a un precio mayor que el de compra). En ese caso la rentabilidad o retorno se calcularía de la siguiente forma:

$$r = (p_t - p_{t-1})/p_{t-1}$$

Donde:

r es la rentabilidad de la inversión, p_{t-1} es el precio de compra y p_t es el precio de venta.

En finanzas cuantitativas en ocasiones se utiliza la rentabilidad logarítmica($R = \ln(p_t/p_{t-1})$) debido a sus propiedades y que en valores muy próximos a 0 su valor es igual a la rentabilidad ordinaria.

El seguimiento de los precios para buscar patrones y la predicción futura de los precios permiten conseguir un beneficio y por tanto son áreas de estudio que han sido tratadas ampliamente.

1.1.2. Creación de carteras

Una cartera es un conjunto de acciones que tiene unos objetivos concretos (de rentabilidad/riesgo) en un horizonte temporal. En (Markowitz, 1952) se propone que un inversor buscará siempre maximizar la rentabilidad minimizando el riesgo en la generación de una cartera. Posteriormente en la literatura se han buscado factores que definiremos como indicadores de rentabilidades comunes entre diferentes activos. El primer modelo que se basa en factores (uno solo en este caso) es el CAPM (Sharpe, 1964), este modelo utiliza como factor β que relaciona la rentabilidad del activo con respecto al mercado. En (Fama and French, 1993) se introduce el famoso modelo de 3 factores que además de tener en cuenta β, añade los factores que tienen el cuenta el tamaño y el crecimiento. Los mismos autores en (Fama and French, 2015) completan el modelo con 2 factores adicionales: rentabilidad y patrones de inversión.

La cantidad de factores investigados y publicados ha aumentado mucho desde la publicación del CAPM y siguen encontrándose nuevos factores. En (Cochrane, 2011) se introduce la expresión *factor zoo* refiriéndose a la inquietud sobre la cantidad de factores y si realmente son nuevas aportaciones o la información de los nuevos descubrimientos en este campo se encuentra ya presente en otros factores previos. Debido a la cantidad de factores, existe trabajos de recopilación y validación como en (Hou et al., 2018) o (Feng et al., 2020) en los que nos apoyaremos para seleccionar los factores para este trabajo.

1.1.3. Aprendizaje automático

El aprendizaje automático es una rama de la inteligencia artificial que mediante técnicas estadísticas permite la creación de sistemas capaces de reconocer e inferir patrones. Este aprendizaje se lleva a cabo a partir de conjuntos de datos de los que es posible extraer los patrones para poder clasificar o predecir nuevos elementos o resultados futuros.

En el campo de los mercados y las finanzas numerosos estudios han sido llevados a cabo. Técnicas de aprendizaje automático han sido utilizadas para mejorar modelos de riesgo (Kakushadze and Yu, 2019), aplicadas al mercado de valores (Patel et al., 2015; Brogaard and Zareei, 2018) y, como serán empleadas en el presente trabajo, junto con modelos multifactor (Sugitomo and Minami, 2018; Turunen, 2019; Li and Zhang, 2019; Fu et al., 2020).

1.2. Motivación del trabajo

La creación de una cartera de inversión óptima es un problema de interés tanto para entidades públicas/privadas como para el inversor minorista. Realizar este proceso de una forma automatizable y replicable a conjuntos de datos de distintos mercados es deseable y se realiza actualmente en fondos de inversión.

En este trabajo abordamos la creación de una cartera partiendo de un conjunto de datos de *Quandl* de donde conseguiremos o crearemos los factores necesarios que serán nuestras características para desarrollar un modelo de aprendizaje automático. Este modelo nos permitirá seleccionar los mejores activos para incluir en la cartera en un estilo de “long only” lo que significa que solo compraremos activos para venderlos a un precio más elevado en el futuro o en caso de que otro activo tenga más potencial. Finalmente contrastaremos los resultados del rendimiento de esta cartera con el índice de referencia que en nuestro caso es el SP500.

1.3. Objetivos del trabajo

Se pretende crear una cartera óptima de activos financieros mediante un modelo de factores fundamentales aplicando técnicas de aprendizaje automático. Además comprobaremos si efectivamente la aplicación del modelo proporciona un beneficio con respecto a los activos considerados como referencia. Para ello, los objetivos concretos que se plantean serán:

- Conseguir los datos y crear los factores
- Implementar un modelo para seleccionar activos

- Verificar la rentabilidad frente al índice

1.4. Planificación del proyecto

En la figura 1.1 se presenta la planificación del proyecto.

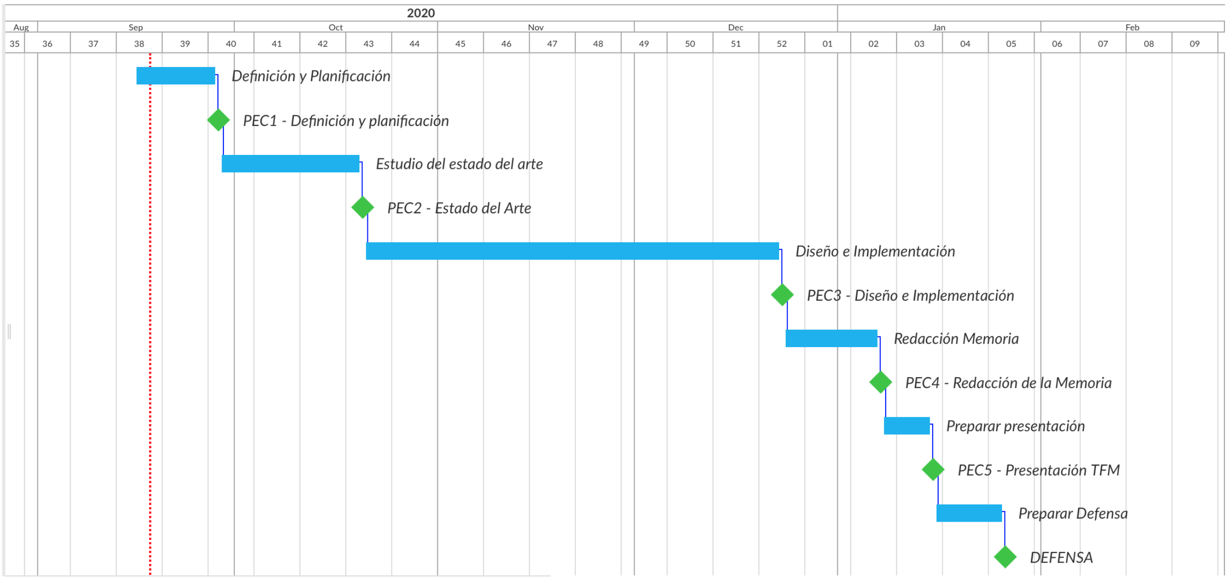

Figura 1.1: Diagrama de Gantt - Planificación del proyecto

1.5. Productos obtenidos

Los productos obtenidos por el trabajo son los siguientes:

1. Repositorio en *Github*, *TFM-UOC-2020* (https://github.com/Caparrini/TFM-UOC-2020.git) (Desde la extracción de los datos hasta las pruebas de la cartera frente a las carteras de referencia).

2. La presente memoria.

1.6. Estructura de la memoria

A continuación se pasa a detallar la disposición de los contenidos de este trabajo:

1. **Introducción**

   Se ofrece un contexto y motivación para el desarrollo de la optimización de cartera de activos financieros. Adicionalmente se comentan los objetivos del trabajo.

2. **Estado del arte**

   En este apartado se trata el estado actual de la optimización de carteras, desde el CAPM hasta los modelos multifactor y la utilización de técnicas de aprendizaje automático en este campo.

3. **Fundamentos conceptuales del trabajo**

   Este capítulo ofrece una visión más específica del proyecto. Se concretan y se definen: Conjunto de datos, procedimientos aplicados, técnicas de aprendizaje automático y el *software* utilizado.

4. **Desarrollo**

   Se exponen los aspectos técnicos del trabajo realizado. Como las acciones seleccionadas para el universo de activos, los factores, los criterios de validación y prueba entre otros.

5. **Resultados**

   Se muestran los resultados y comparaciones con respecto a las cartera modelo.

6. **Conclusiones y trabajo futuro**

   Se expone las conclusiones inferidas de los resultados y posibles líneas de investigación futuras.

Capítulo 2

Estado del Arte

El análisis de carteras y la generación de modelos que estimen el retorno a futuro de los activos ha sido ampliamente estudiado. El que podemos considerar el primer modelo (Sharpe, 1964) utilizaba la β de mercado como su único factor. Posteriormente se publicaría el modelo de 3 factores (Fama and French, 1993) que ha sido punto de partida para otros muchos estudios dentro del ámbito de la búsqueda de "anomalías" (nombre por el que se conocen los factores que tienen capacidad explicativa del retorno de activos frente a otros).

La investigación en este campo ha descubierto numerosas anomalías (314 según (Harvey et al., 2015)) y la mayoría han sido publicadas en las dos últimas décadas. Adicionalmente, la democratización de los datos financieros y su cantidad permite la modelización mediante las técnicas de aprendizaje automático. A su vez, los avances en potencia computacional (tanto *hardware* como *software*) posibilitan la generación de modelos más complejos que puedan aprovechar la cantidad de datos en tiempos más reducidos posibilitando utilizarlos para estrategias de selección de carteras.

2.1. Creación de carteras

En este trabajo nos centramos en modelos multifactor en los que, además, se emplean técnicas de aprendizaje automático. Sin embargo, en primer lugar comentaremos teorías de carteras que han evolucionado en las soluciones actuales.

2.1.1. Markowitz

Una de las primeras teorías de carteras asume que los inversores son reacios a asumir riesgo. En caso de que dos activos presenten la misma rentabilidad elegiremos el que tenga menor riesgo, de esta manera teniendo un universo de activos se puede crear sintéticamente la mejor cartera asignando pesos a los activos de manera que se minimiza el riesgo maximizando la rentabilidad (Markowitz, 1952). Las métricas utilizadas son:

- Retorno esperado (*Expected Return*):

$$E(R_p) = \sum_i E(R_i)$$

Donde R_p es la rentabilidad de la cartera, R_i es el retorno en el activo i y w_i es el peso de la componente i en la cartera.

- Varianza del retorno de la cartera (*Portfolio Return Variance*):

$$\sigma_p^2 = \sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \rho_{ij}$$

donde $\rho_{ij} = 1$ si $i = j$ o:

$$\sigma_p^2 = \sum_i \sum_j w_i w_j \sigma_i \sigma_j \sigma_{ij}$$

donde $\sigma_{ij} = \sigma_i \sigma_j \rho_{ij}$ es la covarianza de la muestra de los retornos de los dos activos que puede representarse como $\sigma(i, j)$, cov_{ij} o $cov(i, j)$.

- Volatilidad del retorno de la cartera (desviación estándar):

$$\sigma_p = \sqrt{\sigma_p^2}$$

Esta teoría se engloba dentro de las consideradas *mean-varianze* que quiere decir que compara la media esperada de los retornos de la cartera con la varianza de esa misma cartera. Considerando un universo de activos y dibujando en un gráfico donde en la x representamos la volatilidad y en la y el retorno tenemos lo que se denomina *frontera eficiente* que son los activos que tienen más rentabilidad en el mismo nivel de riesgo como se puede observar en al figura 2.1.

Este método tiene algunos inconvenientes. Puede producir problemas computacionales cuando el universo de activos que se tiene en cuenta es grande y como variable explicativa únicamente se considera el precio (media y desviación estándar).

2.1.2. CAPM

En primer modelo utilizaba un solo factor (Sharpe, 1964). El el CAPM (por las siglas en inglés de *Capital Asset Pricing Model* el factor utilizado (β) indica el riesgo adicional con respecto al mercado que tiene un activo en concreto. La β puede calcularse utilizando

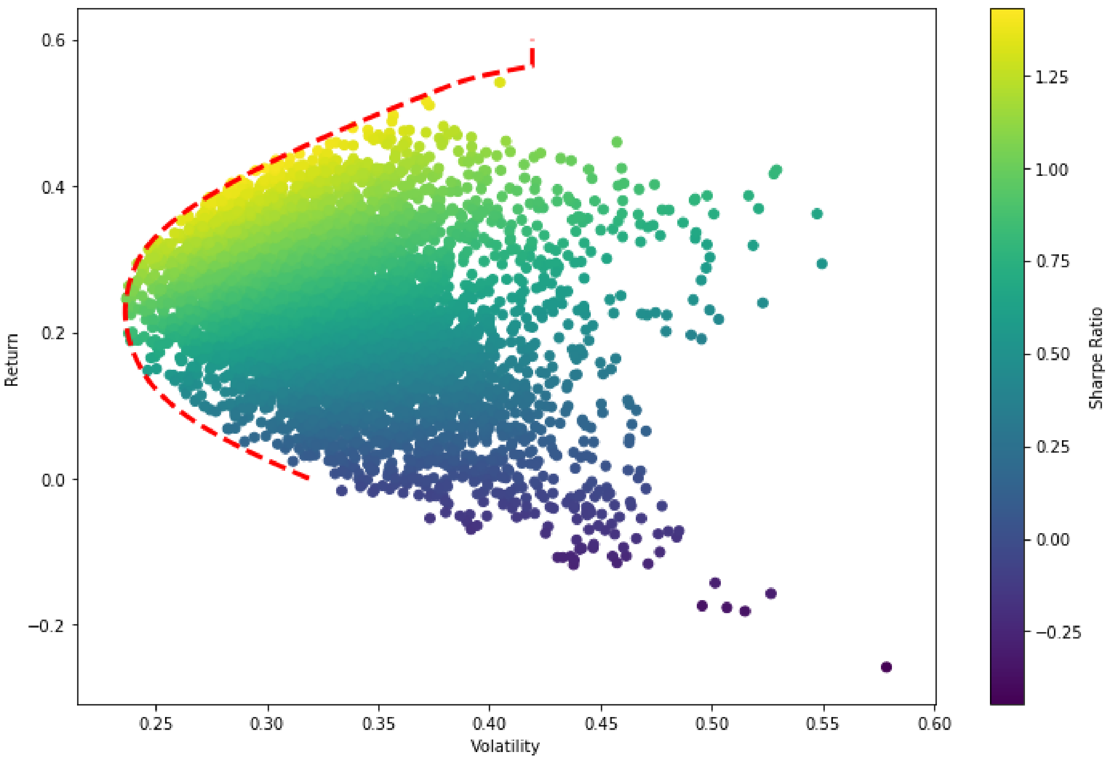

Figura 2.1: Frontera eficiente

regresiones lineales, las covarianzas/varianzas o utilizar la publicada por algún investigador[1].

$$ER_i = R_f + \beta_i(ER_m - R_f)$$

Donde:

ER_i = El retorno esperado del activo i

R_f = La tasa de retorno libre de riesgo (que suele asociarse a la de un bono gubernamental)

β_i = beta de la inversión

$(ER_m - R_f)$ = prima con respecto al riesgo de mercado

Por ello tenemos que cuando $\beta > 1$ el activo tiene más riesgo que el mercado en su conjunto y por tanto debe dar mayor rentabilidad, y si $\beta < 1$ el activo tiene menos riesgo que el mercado y la rentabilidad esperada puede ser menor.

[1]Las Betas publicadas por Aswath Damodaran son ampliamente reconocidas http://pages.stern.nyu.edu/~adamodar/New_Home_Page/datafile/Betas.html

2.1.3. *Fama and French* modelo de 3 factores

En el modelo de Fama y French (Fama and French, 1993) se utilizan 3 factores. Además de la hipótesis de que una acción con más riesgo que el mercado proporcionará mas beneficios (β) se añade una componente sobre el tamaño (empresas más pequeñas tienen más potencial de crecimiento) y otra sobre el valor (empresas con un valor de mercado menor al de los balances de cuentas tienen más potencial de aumentar su valoración).

$$R_{it} - R_{ft} = \alpha_{it} + \beta_1(R_{Mt} - R_{ft}) + \beta_2 SMB_t + \beta_3 HML_t + \epsilon_{it}$$

Donde:

Rit = el retorno total del activo i en el instante t

R_{ft} = la tasa de retorno libre de riesgo en el instante t

R_{Mt} = el retorno de la cartera total del mercado en el instante t

$R_{it} - R_{ft}$ = exceso de retorno sobre la cartera de mercado

SMB_t = prima sobre el tamaño

HML_t = prima sobre el valor

$\beta_{1,2,3}$ = coeficientes de los factores

2.1.4. *Fama and French* modelo de 5 factores

Posteriormente Fama y French (Fama and French, 2015) ampliaron a un modelo de 5 factores. Adicionalmente a los presentes en el modelo de 3 factores tenemos uno sobre los beneficios y otro sobre la inversión.

$$R_{it} - R_{Ft} = \alpha_i + \beta_i(R_{Mt} - R_{Ft}) + \sigma_i SMB_t + h_i HML_t + r_i RMW_t + c_i CMA_t + e_{it}$$

Donde:

R_{it} = retorno del activo i en el instante t

R_{Ft} = tasa de retorno libre de riesgo

$R_m - R_f$ = exceso de retorno sobre la cartera de mercado

SMB = prima sobre el tamaño

HML = prima sobre el valor

RMW = prima sobre los beneficios

CMA = prima sobre es estilo de inversión

2.1.5. *Factor Zoo*

Numerosos estudios han localizado diferentes factores (o "anomalías") que consideran responsables de los retornos de los activos. Cientos de candidatos han sido publicados como fue indicado inicialmente en (Cochrane, 2011) que introdujo el concepto de *Factor Zoo* refiriéndose a la cantidad de factores que surgían y su dudosa validez. Se cuestiona en la literatura si realmente los factores que están apareciendo aportan más información o la información que aportan ya está contenida en otros factores como por ejemplo en (Harvey et al., 2015), (Hou et al., 2018) y (Feng et al., 2020).

Una vez un factor es publicado tiende a disiparse su efecto. Este comportamiento es debido a que si la anomalía se conoce los agentes interesados invertirán en él empujando los precios hasta la corrección. Este comportamiento ha sido estudiado en trabajos como (Mclean and Pontiff, 2016) donde tratan este fenómeno en el mercado de los Estados Unidos.

2.2. Inversión por factores y aprendizaje automático

En este trabajo pretendemos producir una síntesis de los factores que permita detectar los mejores activos. Este problema ha sido abordado a través del aprendizaje automático, tanto no supervisado como supervisado, aunque en este trabajo nos centraremos en el último (más detalles en la sección 3.3). Independientemente de las características y el ámbito de aplicación, las técnicas de aprendizaje automático en finanzas ha explotado durante los últimos años. En el campo de la selección de carteras inicialmente utilizando únicamente datos de precios pero recientemente con conjuntos más completos con características más complejas. Entre los trabajos que utilizan aprendizaje automático supervisado en un modelo multifactor para la selección de activos actualmente podemos encontrar autores que proponen el uso de XGBoost con factores (Jidong and Ran, 2018), conjuntos de técnicas (redes neuronales, técnicas arbóreas y SVM[2]) (Sugitomo and Minami, 2018; Turunen, 2019; Rasekhschaffe and Jones, 2019) o sólo técnicas arbóreas (Fu et al., 2020). Es planteado como trabajo futuro en algunos de ellos la optimización de hiperparámetros lo cuál será el aporte de valor del presente proyecto.

En este trabajo se generaran modelos con XGBoost (Chen and Guestrin, 2016) optimizando mediante algoritmos genéticos que quedarán detallados en las secciones 3.3.1 y 3.3.2.

[2] *Support Vector Machine*

2.3. Conjuntos de datos

Nunca antes ha existido una democratización de datos financieros como la tenemos hoy. En cualquier campo de estudio los datos de partida son de gran relevancia y deben ser de calidad y confianza. Como proveedores de este tipo de datos encontramos entidades líderes de mercado (ej: Bloomberg, Thomson-Reuters, CRSP, Morningstar) y nuevas fuentes alternativas (ej: Yahoo Finance, AlphaVantage, Quandl). En este trabajo se ha optado por el paquete de datos publicado por **Sharadar** en **Quandl**.

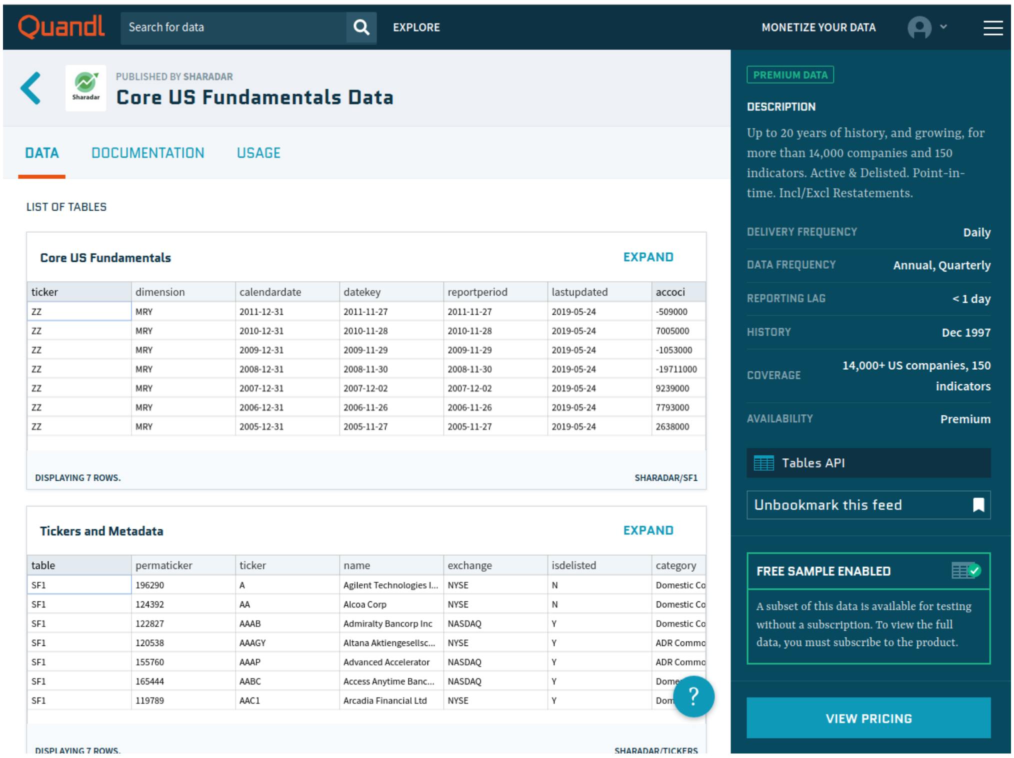

Figura 2.2: Sharadar Core Us Equities Bundle

Sharadar es una firma independiente de análisis e investigación fundada en 2013. Está especializada en la extracción, estandarización y organización de datos financieros a partir de los informes periódicos presentados por las empresas. Combinan personas, *software* y procesos para generar datos de calidad profesional y precisa para inversores profesionales y analistas.

El paquete **Sharadar Core Us Equities Bundle**[3] contiene información financiera de acciones en Estados Unidos tanto de precios y fundamentales de las compañías como de métricas y análisis de terceros.

[3]Más informarción en `https://www.quandl.com/databases/SFA/data`

El conjunto de datos ofrece una licencia específica para la investigación, enseñanza y aprendizaje que permite utilizarlo para el presente trabajo.

2.4. Herramientas software

Se ha utilizado *Python* como lenguaje principal para nuestra aplicación. Este lenguaje permite un "prototipado" rápido y ha sido fuertemente adoptado en el campo de las finanzas por su facilidad de uso y las librerías *Open Source* que existen. Las librerías utilizadas son:

- **Obtención de datos**
    - **Quandl API**
      Quandl proporciona acceso a millones de conjuntos de datos financieros y económicos, algunos son de acceso libre y otros requieren una cuenta de pago. Accesible mediante una API para *Python*.
    - **Zipline**
      Zipline es un librería escrita en *Python* que permite crear sistemas de *trading* algorítmico. Tiene funcionalidades para a partir de los datos generar los factores (predefinidos o personalizados).
- **Aprendizaje automático**
    - **Pandas**
      Pandas es una librería de código abierto, que proporciona estructuras de datos y herramientas de análisis de éstos para *Python*.
    - **Scikit-learn**
      Scikit-learn de código abierto, ofrece herramientas para la minería y el análisis de datos. Contiene implementaciones de algoritmos de aprendizaje automático y herramientas para facilitar su uso y entrenamiento.
    - **DEAP**
      Facilita el "prototipado" rápido de optimizaciones con algoritmos genéticos. Es libre, bien documentado y está implementado en Python.
- **Representación de resultados**
    - **Matplotlib**
      Matplotlib es una biblioteca de *Python* que genera gráficos en 2D. Usado para la representación gráfica de resultados.

Capítulo 3

Fundamentos conceptuales del trabajo

En este capítulo, se muestra el material utilizado así como una explicación descriptiva de los conceptos clave que se han tenido en cuenta en el trabajo.

Para la consecución de los objetivos mencionados en la introducción y la generación de un modelo predictivo para la selección de activos para la cartera el trabajo debe cumplir las siguientes fases:

- Elección universo de acciones.
- Elección de los factores para estos activos.
- Elección de los algoritmos de aprendizaje automático.
- Optimización de los hiperparámetros de los algoritmos.
- Implementación de la generación de los factores y entrenamiento del clasificador.
- Validación de los resultados.

3.1. Conjunto de datos

El conjunto de datos seleccionado para el trabajo es *Core US Fundamental Data* publicado por *SHARADAR* en la plataforma *Quandl*. Este conjunto de datos incluye los siguientes componentes:

- Core US Fundamentals Data
- Core US Insiders Data
- Core US Institutional Investors Data
- Sharadar Equity Prices
- Sharadar Fund Prices

Entre toda esta información tenemos en resumen el histórico de precios de acciones en US así como datos fundamentales de las empresas.

Factor	**Category**
6-month prior return	Momentum
11-month prior return	Momentum
Return on equity	Profitability
beta 3Y covariance	Intangible
beta 3Y linear reg	Intangible
Price-to-book	Value-vs-growth
Earnings-to-price	Value-vs-growth
Price-to-sales	Value-vs-growth
Market capitalization	Trading frictions
Enterprise-value	Value-vs-growth
evebit	Value
evebitda	Value
12 month lagged return	Intangibles
24 month lagged return	Intangibles
Operating cash flow-to-price	Value-vs-growth
Investment-to-price	Investment
Earning per share	Profitability
Current ratio	Intangibles
Operating cashflow-to-equity	Profitability
Capex	Intangibles

Cuadro 3.1: Tabla factores

3.2. Factores

Los factores a utilizar de los diferentes activos van a ser los especificados en la tabla 3.1. Muchos de ellos son los definidos en (Hou et al., 2018) como algunos de los factores más significativos dentro de la literatura. Está fuera del alcance de este proyecto el análisis, búsqueda o comparación de todos los factores que han sido investigados en otros trabajos.

3.2.1. Universo de activos a considerar

Los activos que consideramos dentro de nuestro universo de prueba son todas las empresas que han pertenecido al SP500 desde inicios desde el año 1997 hasta 2020. Nuestro objetivo es en cada instante del tiempo que seleccionemos activos, seleccionar un subconjunto muy reducido (15 activos) del SP500 de manera que sean los que estimamos tengan una mejor rentabilidad. Es importante recalcar que en cada fecha que se realice una selección solo se tendrán en cuenta las empresas que conforman el SP500 en esa misma fecha, y no todas las que han formado parte del índice hasta ahora.

3.2.2. Proceso de generación de los factores

Algunos de los factores utilizados son una métrica proporcionada directamente por el proveedor del conjunto de datos, otras son calculados usando una combinación de métricas o el histórico de precios.

3.2.3. Definición de los factores

Una explicación de cada factor tenido en cuenta en la tabla 3.1:

- **6-month prior return**

  Métrica que indica la rentabilidad acumulada del activo durante los últimos 6 meses. Es una métrica relacionada con el "momentum" cuya hipótesis es que los activos que han tenido mejor desempeño lo mantienen.

  $$r_{6m} = p_0/p_{-1}$$

  Siendo:

  p_0 el precio en el instante actual t_0

  p_{-1} el precio en $t_0 - 6$ meses

- **11-month prior return**

  También relacionada con el "momentum". Indica la rentabilidad acumulada del activo durante los últimos 11 meses.

  $$r_{11m} = p_0/p_{-1}$$

  Siendo:

  p_0 el precio en el instante actual t_0

  p_{-1} el precio en $t_0 - 11$ meses

- **Return on equity**

  Es una métrica del rendimiento financiero que se calcula dividiendo los beneficios netos entre el patrimonio neto. Es un medida de lo rentable que es un negocio en relación a los accionistas.

  En el conjunto de datos de fundamentales SF1 tenemos el campo calculado en la etiqueta "roe".

  $$ReturnonEquity = \frac{NetIncome}{AverageShareholders'Equity}$$

- **β 3Y covariance**

  β es la sensibilidad del precio de un activo al mercado en su conjunto. Es una medida de la volatilidad o el riesgo. Mayores valores implican que en caso de un movimiento del mercado (en cualquier dirección) implican mayores movimientos de la acción en la misma dirección. Para calcularla usamos el histórico de precios durante 3 años del SP500 y de cada acción de la que queremos extraer la β mediante la siguiente fórmula.

  $$\beta_s = \frac{Cov(r_s, r_b)}{Var(r_b)}$$

  Siendo:

  β_s la β del activo con respecto al mercado o referencia

  r_s las rentabilidades diarias para el activo del que queremos extraer la β

  r_b las rentabilidades diarias del mercado o referencia ("benchmark") en nuestro caso el SP500

- **β 3Y regresión linear**

  β también puede calcularse utilizando una regresión lineal. La regresión se entrena con las rentabilidades del activo y se estima el valor del mercado o referencia (SP500), de manera que el coeficiente de la regresión es la β.

- **Price-to-book**

  Este ratio se utiliza para comparar el precio de mercado de la compañía contra su valor en libros.

  En el conjunto de datos de fundamentales SF1 tenemos el campo calculado en la etiqueta "pb".

  $$Pricetobook = \frac{CommonShareholders'Equity}{MarketCap}$$

- **Earnings-to-price**

  Este ratio relaciona el precio de las acciones de una compañía respecto a las ganancias por acción.

  En el conjunto de datos de fundamentales SF1 tenemos el campo calculado en la etiqueta "pe".

  $$Earnings - to - price = \frac{Anualearnings}{Priceofsecurity}$$

- **Market capitalization**

  Representa el valor total de mercado de la compañía utilizando el precio de la acción y la cantidad de acciones.

  En el conjunto de datos de fundamentales SF1 tenemos el campo calculado en la etiqueta “marketcap”.

- **Enterprise-value**

  Es una medida del negocio en su conjunto. Calculado como la capitalización de mercado más la deuda menos el efectivo y otros activos líquidos equivalentes.

  En el conjunto de datos de fundamentales SF1 tenemos el campo calculado en la etiqueta “ev”.

- **evebit**

  Ratio que relaciona el valor del negocio con el EBIT[1].

  En el conjunto de datos de fundamentales SF1 tenemos el campo calculado en la etiqueta “evebit”.

- **evebitda**

  Ratio que relaciona el valor del negocio con el EBITDA[2].

  En el conjunto de datos de fundamentales SF1 tenemos el campo calculado en la etiqueta “evebitda”.

- **12 month lagged return**

  Los retornos con “lag” son el “momentum” que tenía en el pasado. En este caso el retorno con “lag” de 12 meses es considerando el punto inicial de hace 12 meses el retorno acumulado de los últimos 12 meses.

- **24 month lagged return**

  En este caso el retorno con “lag” de 24 meses es considerando el punto inicial de hace 24 meses el retorno acumulado de los últimos 12 meses.

- **Operating cash flow-to-price**

  Es el ratio entre los flujos de entrada de efectivo por operaciones con respecto a la capitalización de mercado.

- **Investment-to-price**

  Inversiones son los activos financieros de la compañía. Este ratio relaciona estas inversiones en el balance con respecto a la capitalización de mercado de la compañía.

[1] *Earnings before interest and taxes*

[2] *Earnings before interest, taxes, depreciation and amortization*

- **Earnings per share**

  Ganancias por acción es el beneficio neto dividido entre la cantidad de acciones.

  En el conjunto de datos de fundamentales SF1 tenemos el campo calculado en la etiqueta "eps".

- **Current ratio**

  El ratio entre activos corrientes y pasivos corrientes.

  En el conjunto de datos de fundamentales SF1 tenemos el campo calculado en la etiqueta "currentratio".

- **Operating cashflow-to-equity**

  Es el ratio entre los flujos de entrada de efectivo por operaciones con respecto al patrimonio neto de la compañía.

- **Capex**

  Son los fondos usados por una compañía para adquirir, mejorar o mantener activos físicos como edificios, fábricas, equipamiento o tecnología.

  En el conjunto de datos de fundamentales SF1 tenemos el campo calculado en la etiqueta "capex".

3.3. Aprendizaje automático

El aprendizaje automático busca extraer patrones o información a partir de datos. Estás técnicas pueden clasificarse dentro de dos grupos en función de si los datos de los que partimos están o no etiquetados:

1. **Aprendizaje supervisado**

   Los modelos parten de conjuntos de datos que están previamente etiquetados. El modelo se entrena con los datos e infiere patrones para poder asignar una clase a nuevas muestras de datos sin etiquetar. Este es el tipo de aprendizaje automático que vamos a utilizar siendo nuestro objetivo final la predicción de una etiqueta (clase) para cada activo financiero.

2. **Aprendizaje no supervisado**

   Este tipo parte de datos que no se encuentra previamente etiquetada. Mediante estas técnicas los datos se agrupan en algunos conjuntos en función de la similitud en sus características, siendo los elementos que pertenecen a un mismo grupo más parecidos entre ellos que los que se encuentran en grupos distintos.

Los algoritmos que se utilizan en este trabajo son de aprendizaje supervisado y la técnicas en concreto se detallan en los siguientes apartados.

3.3.1. Árboles de decisión

Utilizamos árboles de decisión[3] ya que son simples y rápidos además de permitir extraer la importancia de las características. Son modelos de los que podríamos considerar "white-box" debido a que se puede extraer e incluso visualizar como el algoritmo a tomado una decisión de clasificación para un caso en concreto.

Un árbol de decisión proporciona un método de clasificación que construye un modelo a partir de un conjunto de datos debidamente etiquetado con la clase a la que corresponden para posteriormente, clasificar nuevos datos desconocidos (Breiman et al., 1984).

Una vez el árbol está entrenado contiene reglas que a partir de las características de un nuevo individuo le asignarán la clase. Podemos ver un ejemplo en la figura 3.1 de un pequeño árbol de decisión.

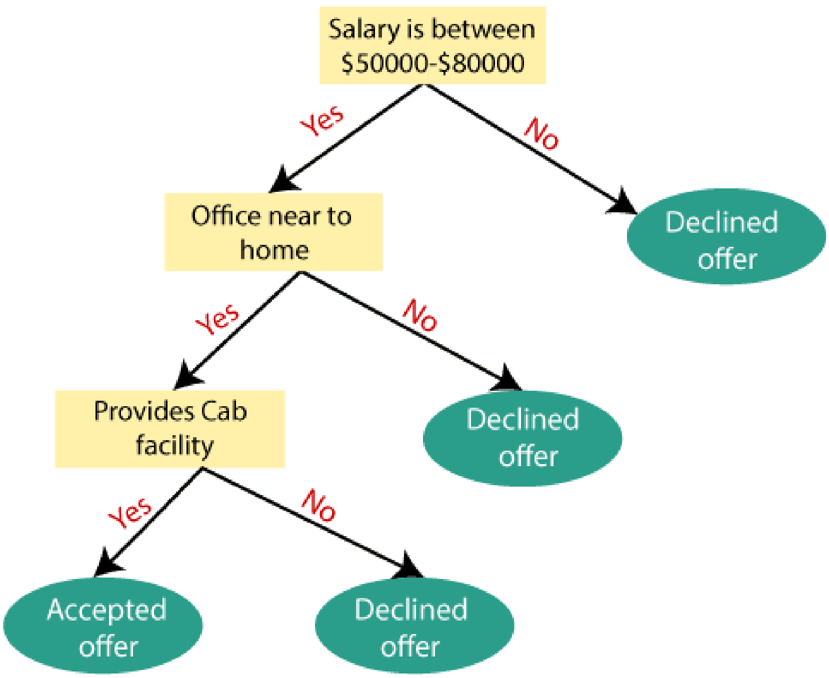

Figura 3.1: Ejemplo de árbol de decisión

[3]Referencia: http://scikit-learn.org/stable/modules/tree.html

3.3.2. Bosques aleatorios

Para complementar los resultados y lograr una precisión mayor de la clasificación utilizamos otro algoritmo en este trabajo: el bosque aleatorio[4].

El bosque aleatorio[5] es una agregación de árboles de decisión. Las agregaciones de clasificadores son utilizadas ya que normalmente funcionan bien debido a que cada clasificador aprende en detalle diferentes aspectos del conjunto de datos y luego ponen en común las predicciones teniéndose en cuenta la mejor o más repetida. En algoritmos de agregación tenemos métodos de "*bagging*" que se basan en entrenar diferentes instancias de algoritmos (como en este caso árboles) con partes aleatorias del conjunto de datos para entrenar. La forma de elegir estas partes aleatorias depende de los distintos agregados, en el caso del bosque aleatorio utilizamos la **selección con reemplazamiento**, que siendo n el conjunto de datos para entrenar coge n muestras y en cada uno tiene una probabilidad de $(1-\frac{1}{n})^n$ de ser omitido (ya que es elegido varias veces).

El objetivo de agregar distintos árboles entonces es construir un modelo que mejora la generalización y robustez frente a un solo árbol. Para ellos construimos los árboles independientemente y luego promediamos las predicciones para dar la predicción final del bosque.

3.3.3. XGBoost

El algoritmo eXtreme Gradient Boosting (XGBoost)(Chen and Guestrin, 2016) es una implementación eficiente del "gradient boosting"[6]. Este método ha crecido en popularidad ya que ha ganado varias competiciones y numerosas investigaciones lo utilizan.

Tiene las ventajas de los bosques aleatorios mencionadas anteriormente con el añadido de que mejore la tasa de aciertos mediante la función de gradiente de los predictores débiles.

Importancia de las variables

En principio, las variables no utilizadas en un árbol de decisión no son importantes, aunque puede no ser cierto en algunos casos, como en el caso en el que sea una variable correlacionada con otra o redundante. La importancia de las variables utilizadas en un árbol se pueden medir de forma individual utilizando la impureza Gini. Cada vez que partimos un nodo con una variable resultan dos nodos hijos con menos impureza. La importancia de la variable la conseguimos agregando el decrecimiento que se produce en la impureza al

[4]Referencia: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[5]Página de Breiman donde explica los bosques aleatorios: https://www.stat.berkeley.edu/ breiman/RandomForests/cc_home.htm
[6]https://xgboost.readthedocs.io/en/latest/

partir para esa variable a lo largo del árbol cuando haya sido utilizada.

En los bosques aleatorios, que están formado por un conjunto de árboles de decisión, se puede medir la importancia de las variables de cada árbol, de manera que una característica es importante a partir de cómo decrece la impureza del árbol. Este decrecimiento de la impureza se promedia para todos los árboles del bosque y se consigue la importancia en el clasificador global. Este método se denomina como la media del decrecimiento de la impureza (*mean decrease impurity*)[7].

3.4. Validación de la clasificación

Para generar los hiperparámetros de los algoritmos se ha utilizado una implementación de un algoritmo genético (explicado durante el desarrollo en la sección (**?**)) donde la métrica a optimizar ha sido la tasa de aciertos balanceada por clases. Para calcular esta métrica se calcula utilizando una validación cruzada para series temporales de $k = 5$ particiones.

3.4.1. Validación cruzada de K iteraciones para serie temporal

Disponiendo de un conjunto de datos surge la inquietud de como optimizar el algoritmo sin incurrir en sobre-entrenamiento. En el caso de partir el conjunto en un conjunto para entrenamiento y otro para validación, nos encontramos con la problemática de que podemos estar capturando muy específicamente el conjunto de validación, evitando que el algoritmo generalice correctamente a partir de los datos.

Una de las técnicas más ampliamente utilizadas es la validación cruzada. El conjunto de datos se divide en k partes, utilizándose $k - 1$ partes para entrenar y la restante para la validación. Este proceso se puede iterar k veces, obteniendo k valores para la tasa de aciertos. Este procedimiento con series temporales requiere de ciertas modificaciones. Cuando utilizamos información temporal hay que garantizar que el algoritmo no tiene acceso a la información futura que no debería conocer en el instante de la prueba. El procedimiento (como se indica en la figura 3.2) es el siguiente:

1. Dividimos el conjunto inicial en k subconjuntos de manera ordenada en el tiempo.

2. Para cada i perteneciente a $(0, k - 2)$ utilizamos todos las particiones hasta $k = i$ como entrenamiento, evitamos la partición $k = i+1$ y utilizamos la partición $k = i+2$ como conjunto de test. Recalcar que evitamos una partición para evitar "look-ahead

[7] Más información y formulación matemática en el capítulo 5.3.4 del libro escrito por Breiman et al. (1984) que es la fuente que inspira la implementación en *scikit learn*

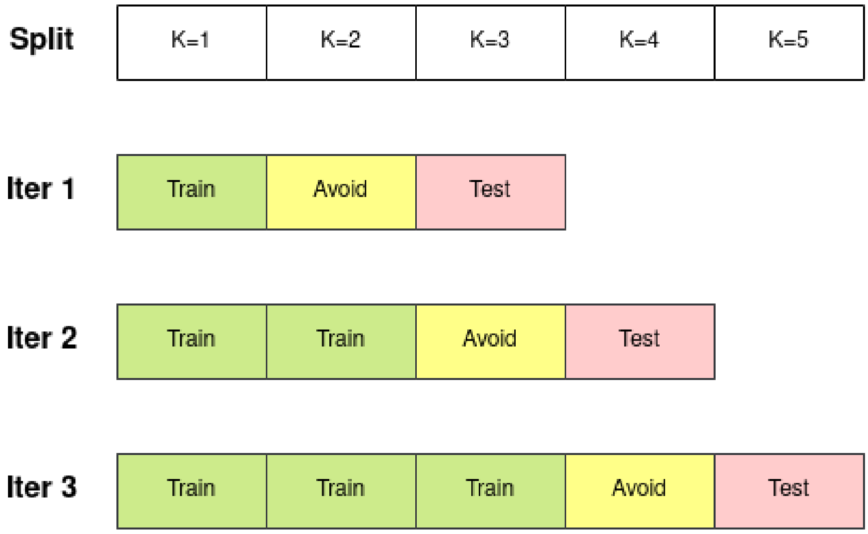

Figura 3.2: Validación temporal cruzada de K iteraciones

bias”[8] y que en concreto evitamos una parte ya que el conjunto con el que generamos el clasificador optimizado.

3. Una vez realizado el proceso ($k-2$ veces) calculamos la media de las tasas de aciertos balanceadas de las diferentes iteraciones obteniendo una métrica agregada.

En nuestro caso hemos empleado esta técnica utilizando un $k = 5$, de manera que hay 3 iteraciones. Teniendo en cuenta la naturaleza de los datos, la etiqueta que pretendemos estimar está basada en el retorno de un activo en los próximos 11 meses. Para evitar un sesgo del algoritmo de conocimiento del futuro y sobre-entrenamiento eliminados la partición temporalmente posterior en cada iteración.

[8]Es un tipo de sesgo que ocurre cuando en una simulación se utilizan datos que realmente no serían conocidos durante el periodo estudiado.

Capítulo 4

Desarrollo

En este apartado se detallan los pasos llevados a cabo para la consecución de los objetivos del trabajo, es decir, probar y desarrollar la estrategia propuesta. El código utilizado para el desarrollo se encuentra en *Github*, *TFM-UOC-2020* (https://github.com/Caparrini/TFM-UOC-2020.git). A continuación se indican las 3 secciones de este capítulo:

1. **Generación de conjuntos de datos**

   En primer lugar formamos el conjunto de factores conforme a lo definido en la sección 3.1.

2. **Generación de mejores parámetros**

   Para cada uno de los tres algoritmos utilizados (árbol de decisión, bosque aleatorio y XGBoost) realizamos una optimización de los hiperparámetros que serán los parámetros utilizados en cada una de los backtest.

3. **Backtest**

   Definimos el funcionamiento de la estrategia y como se realiza. Esta estrategia se lleva a cabo para cada uno de los 3 algoritmos optimizados en el paso anterior.

4.1. Conjunto de datos del trabajo

El conjunto de datos está formado por 20 características de todas las empresas que han pertenecido al SP500 desde el año 1997 hasta el 2019. Estás características están definidas en las sección 3.1. Los procesamientos realizados en algunas de ellas, así como el cálculo de las betas se encuentran en el repositorio en *Github*, *TFM-UOC-2020* (https://github.com/Caparrini/TFM-UOC-2020.git), concretamente, en el “notebook” **TFM - Data**.

Para la variable objetivo (o clase) que vamos a predecir con los modelos de aprendizaje automático supervisado utilizamos las de la tabla 4.1 a partir de los retornos del activo durante los próximos 12 meses. Esta forma de etiquetar es similar a la utilizada en (Fu et al., 2020), sin embargo, en nuestro caso utilizamos el retorno anual en lugar del mensual.

Retorno 12 meses	**Clase**
$retorno >= 0{,}15$	4
$0{,}05 <= retorno < 0{,}15$	3
$0 <= retorno < 0{,}05$	2
$-0{,}15 <= retorno < 0$	1
$retorno < -0{,}15$	0

Cuadro 4.1: Clase

4.2. Generación de mejores parámetros

Para la optimización de igual forma que en (Caparrini López and Pérez Molina, 2017) utilizamos la librería **DEAP** para *Python* que permite rápida prototipación de algoritmos genéticos parametrizando particularmente lo que el interesado considera individuos y la métrica de evaluación de individuos a optimizar. Un algoritmo genético es un algoritmo de búsqueda del mejor resultado basado en las ideas de la selección natural y la genética. De esta forma, diferentes soluciones al problema, son generadas de forma aleatoria al principio. Esta sería la primera generación, y las mejores soluciones, comparten parámetros entre ellas formando nuevas generaciones. Además en las diferentes generaciones se introducen mutaciones para generar aleatoriedad. De esta forma, vamos consiguiendo cada vez mejores resultados, y aunque no garantiza el mejor resultado, proporciona un resultado bueno ahorrando computación y recursos[1].

Para nuestras optimizaciones un individuo consistiría en el conjunto de parámetros de un clasificador y la evaluación del individuo es la tasa de aciertos balanceada calculada mediante una validación cruzada como se especifica en 3.4.1.

Los parámetros definidos para la optimización son los siguientes:

- Generaciones: 10 (4 XGBoost)
- Individuos iniciales: 30 (13 XGBoost)
- Forma de cruzar: two-point crossover
- Selección: Seleccionar el mejor individuo de entre 4 seleccionados aleatoriamente, repetido 30 veces.
- Mutación: Mutación de un individuo sustituyendo atributos, con una probabilidad de 0.35 generando un valor aleatorio entre unos umbrales mínimo y máximo definidos.

Con estos parámetros para el algoritmo genético realizamos una optimización para los algoritmos para la primera fecha de la estrategia t_0.

[1] Más información en https://www.doc.ic.ac.uk/ nd/surprise_96/journal/vol1/hmw/article1.html

- Optimización de árbol de decisión
- Optimización de bosque aleatorio
- Optimización de XGBoost

4.3. Backtest

En esta sección introducimos el método utilizado para medir la efectividad de los algoritmos en la selección de activos. Esta metodología tiene en cuenta las prácticas que consideramos reducen el sobre-entrenamiento y evitan un sesgo de ofrecer datos futuros a los algoritmos.

En (Arnott et al., 2018) los autores advierten sobre el uso del aprendizaje automático en finanzas. Una gran cantidad de datos permite múltiples capas de validación cruzada pero en finanzas los datos son menores y el ratio de información-ruido es bajo. Además, (Fabozzi and De Prado, 2018) remarca como solo las estrategias que tienen buenos rendimientos llegan a publicarse siendo miles las que nunca llegan a conocerse, esto puede llegar a generar decepción cuando se intenta llevar a un sistema de transacciones real que explote la estrategia.

Por ello tratamos de garantizar que en cualquier fecha el modelo solo tiene en cuenta los datos que serían conocidos en ese punto. Adicionalmente, la primera fecha de la estrategia debe tener suficientes datos como para poder entrenar el primer modelo.

Un algoritmo capaz de predecir los mejores activos a lo largo del tiempo perderá capacidad predictiva si los nuevos datos temporales no son usados en el modelo. Los datos más actuales del mercado pueden incluso generar modelos que sean muy distinto, por ello, los datos más recientes son utilizados en cada instante para entrenar el modelo obteniendo mayor capacidad predictiva. Hemos optado por no utilizar todo el histórico desde el origen agregando nueva información, y solo utilizamos la información más próxima al punto de entrenamiento del modelo. Argumentamos que todo el histórico puede ser confuso para el modelo y que los tiempos de entrenamiento aumentan considerablemente.

En una estrategia de transacciones se transforma lo que se denominaría una señal en pesos de los activos en la cartera. Nuestra señal sera la probabilidad dada por el modelo de pertenencia a la clase 4 (retornos $>= 15\,\%$) de un activo, como cantidad de activos siempre seleccionaremos los primeros 15 y para los pesos aplicamos la política de igualdad de pesos (cada uno será un 1/15 de la cartera).

Para describir la estrategia en primer lugar vamos a definir t_i, $\triangle_S$, $\triangle_H$ y $\triangle_L$:

- t_i: Fecha para hacer el rebalanceo de los activos seleccionados con un modelo reentrenado.

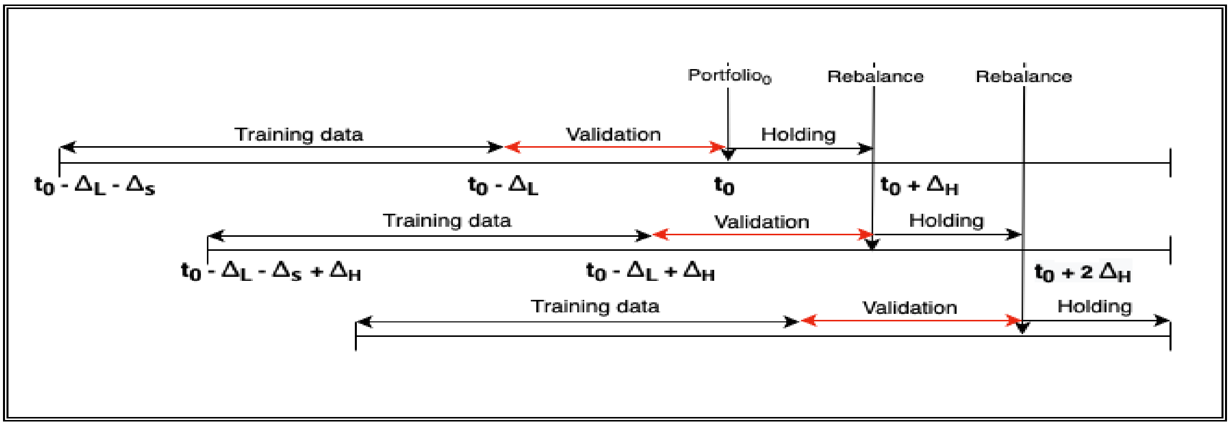

Figura 4.1: Diseño Backtest

- $\triangle_S$: Rango de tiempo del conjunto de datos utilizado para entrenar el modelo (**3 años**)
- $\triangle_H$: Periodo de mantenimiento de los activos en cartera entre fechas de rebalanceo (**4 meses**)
- $\triangle_L$: Periodo de datos no utilizados para evitar sesgos con la variable objetivo (**1 año**)

Con estas definiciones, la estrategia se lleva a cabo siguiendo los siguientes pasos:

- Primero, se selecciona la fecha de rebalanceo (t_i). En esta fecha, un clasificador es entrenado utilizando el rango de datos entre las fechas $(t_i - \triangle_S - \triangle_L, t_i - \triangle_L)$. Evitamos el uso de los datos en el rango de fechas $(t_i - \triangle_L, t_i)$ para evitar sesgo de conocer el futuro.
- Después el clasificador es utilizado para predecir la probabilidad de pertenencia a la clase 4 (retornos $>= 15\,\%$) para los datos en t_i. Estos resultados se usan para generar un ranking de los activos y los 15 primeros son seleccionados para formar parte de la cartera. Los pesos para cada activo son $1/N$ ya que es un sistema simple y difícil de mejorar como se explica en (DeMiguel et al., 2009). Los retornos son calculados para la cartera que se ha seleccionado utilizando los precios de t_{i+1} y t_i.
- A continuación, realizamos el proceso desde el principio ahora en t_{i+1}, siendo $t_{i+1} = t_i + \triangle_H$, repitiendo hasta que alcanzamos la última fecha de rebalanceo.

Para clarificar, en la figura 4.1 mostramos gráficamente estas ventanas en la primera fecha y los dos primeros rebalanceos.

Capítulo 5

Resultados

En este capítulo presentamos los resultados que hemos obtenido a partir de los conjuntos de datos presentados en la sección 4.1, utilizando los procesos de aprendizaje automático explicados en detalle en la sección 4.2. Todas los gráficos presentes en esta sección se han subido en versión "html" a un repositorio en la nube de AWS[1] y son accesibles mediante un hipervínculo asociado a la imagen. De esta forma se puede ver en detalle los valores de una forma interactiva en un navegador. La sección esta organizada de la siguiente forma:

- Árboles de decisión y Bosques aleatorios

  Los resultados de los árboles de decisión y los bosques aleatorios son similares por lo que los vamos a exponer en conjunto. Se muestra por un lado las tasas de aciertos de los algoritmos y por otro las importancias de las características.

- XGBoost

  Los resultados de XGBoost son mejores y los describimos en este apartado, mostrando las tasas de acierto de los distintos clasificadores a lo largo del tiempo y la importancia de las características.

- Resultado final

  En esta sección se muestran los retornos de las estrategias finales y se comparan con respecto al SP500.

5.1. Árbol de decisión y bosque aleatorio

Los bosques aleatorios son técnicas más sencillas y junto con el bosque aleatorio podemos ver en la tabla 5.1 que la tasa de aciertos media ronda el 30 %. Se recuerda que tenemos un clasificador distinto para cada fecha por lo que tenemos distintos clasificadores entrenados en distintas fechas cuyas tasas de acierto se pueden ver en en la imagen 5.5. Ambas tienen una desviación muy similar a lo largo del tiempo.

Estás tasas de acierto son sobre todo el conjunto de pruebas para cada fecha, sin embargo, no muestran el rendimiento de la estrategia de igual forma que los retornos de

[1] *Amazon Web Services*

los activos seleccionados, ya que solo los 15 activos con mayor probabilidad de pertenecer al grupo de mayor rendimiento pronosticado son elegidos para la cartera y serán los únicos que tendrán un efecto sobre el resultado. Los resultados finales de beneficios de la estrategia se muestran en la sección 5.3.

5.1.1. Importancia de las características

En esta sección mostramos la importancia de las características del árbol de decisión y el bosque aleatorio. En las figuras 5.1 y 5.2 podemos ver respectivamente el ranking medio de las características utilizadas por el árbol de decisión y el bosque aleatorio en todos los clasificadores a lo largo del tiempo. También se muestra la desviación estándar para reflejar los cambios a lo largo del tiempo en la importancia de las características.

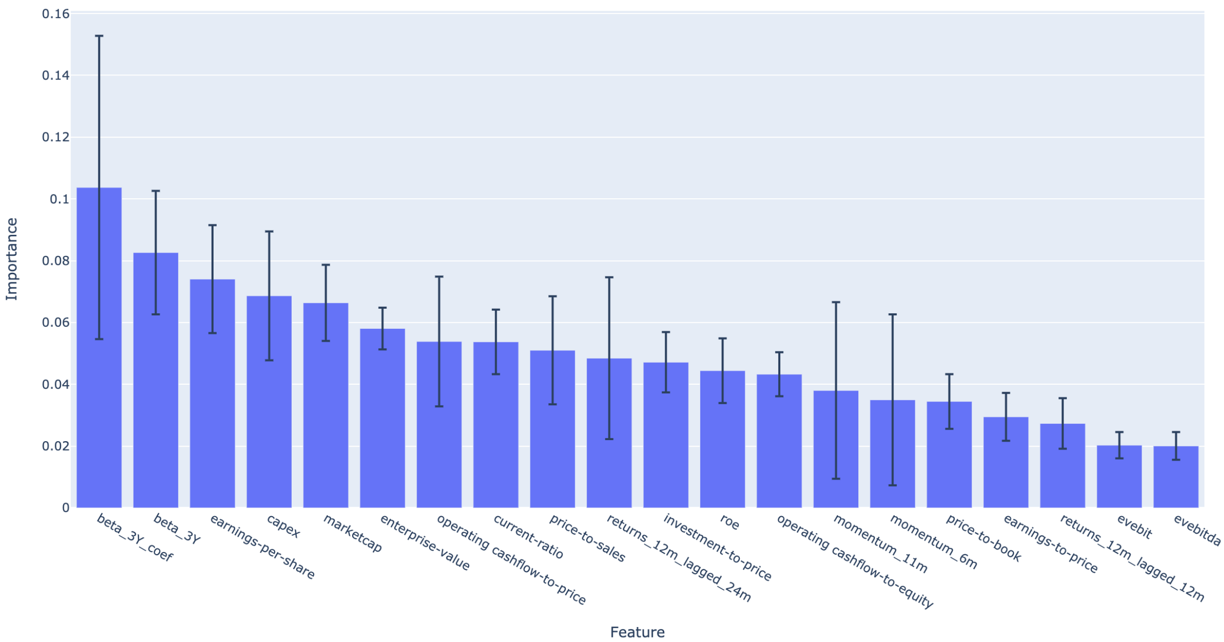

Figura 5.1: Ranking importancia de las características - Árbol de decisión

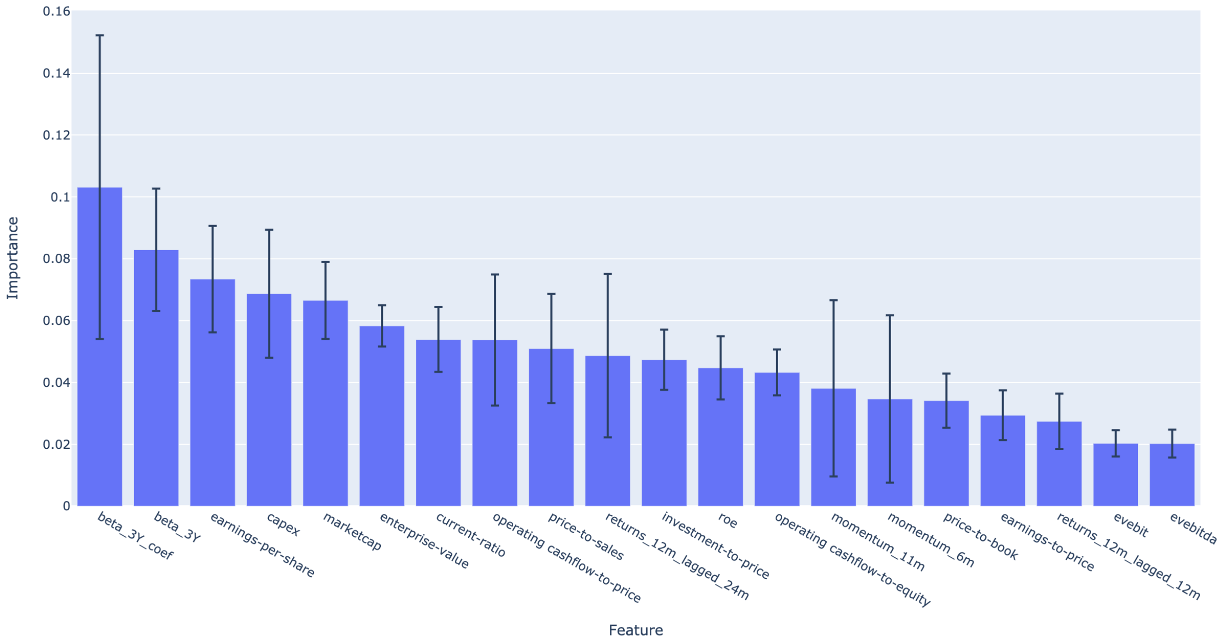

Figura 5.2: Ranking importancia de las características - Árbol de decisión

Dadas estas grandes desviaciones en la importancia de las características también mostramos en las figuras 5.3 y 5.4 la evolución de cada característica individual a lo largo del tiempo.

En estos gráficos se observa un comportamiento interesante que tiene relevancia a la hora de aplicar modelos a la selección de activos para una cartera:

- Características que en un momento son de las más importantes cambian drásticamente en otra fecha. Este comportamiento parece razonable debido al diferente contexto económico en cada instante del tiempo y que el algoritmo es capaz de capturar estas condiciones.

- Se aprecia un cambio en las importancias entre el 2010 y el 2015 coincidiendo con la crisis de 2008 lo cuál es indicativo de que los factores que predicen buenos rendimientos antes y después de una crisis son diferentes.

- Las dos características de media más relevantes son las diferentes betas de mercado. Esto tiene sentido al ser modelos más sencillos y ya que a mayor valor de la beta implica más riesgo pero más beneficio en la misma dirección que el mercado. A pesar de ello, estas características pierden bastante peso tras la crisis, donde otras variables relativas al valor de una compañía cobran especial relevancia (evebit, evebitda, earnings-to-price por nombrar algunas).

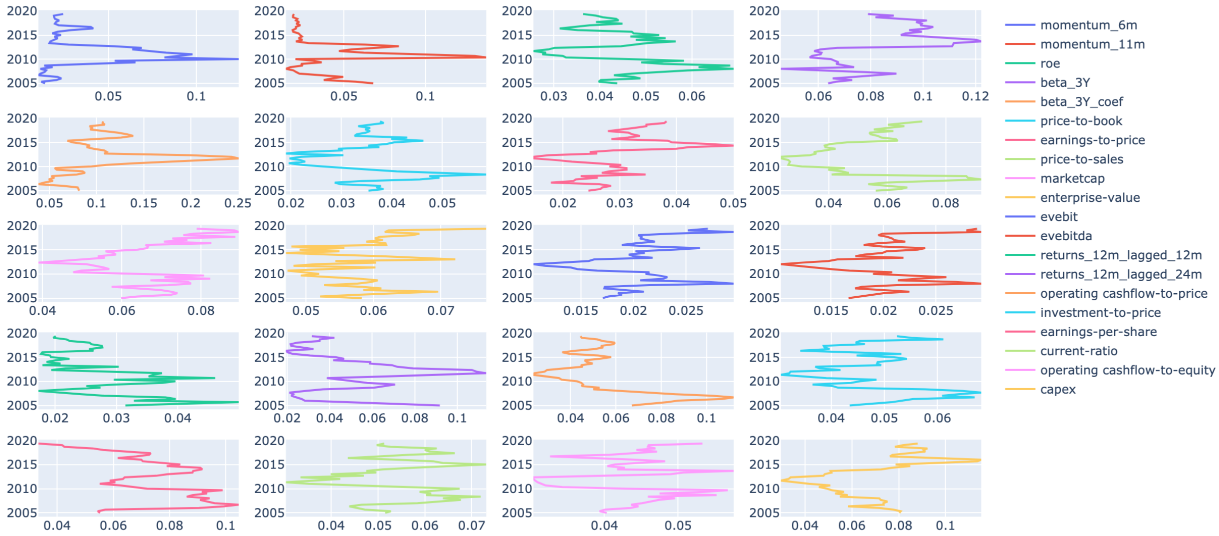

Figura 5.3: Importancia de las características - Árbol de decisión

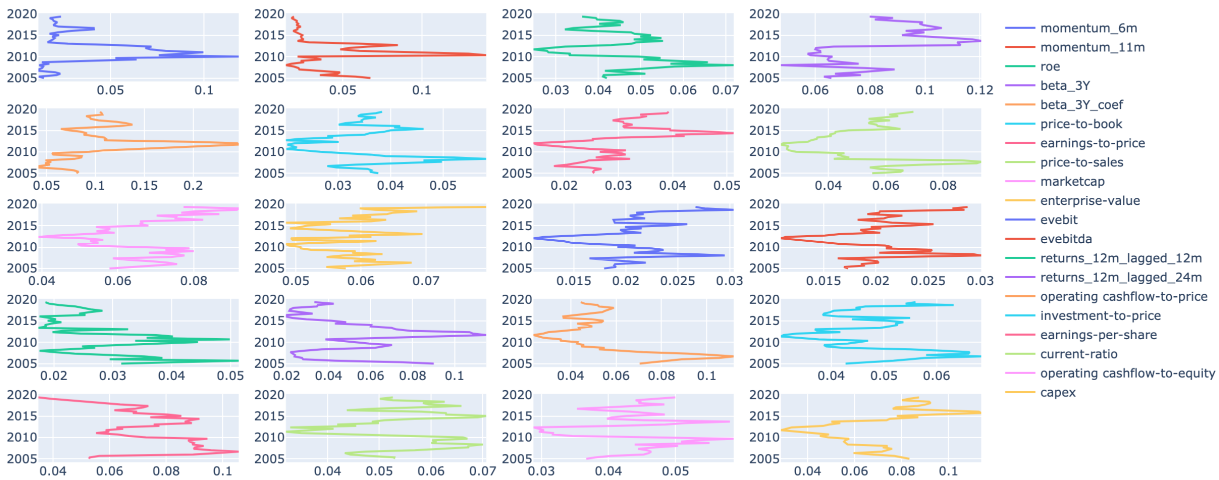

Figura 5.4: Importancia de las características - Bosque aleatorio

5.2. XGBoost

El algoritmo XGBoost tiene una tasa de aciertos promedio del 40 % (tabla 5.1) y aunque la desviación es bastante mayor los resultados que produce la estrategia son bastante buenos. Remarcar como en la sección del dt[2] y rf[3] que la tasa de aciertos es global y aunque

[2] *Decision tree*
[3] *Random forest*

un 40 % sobre 4 clases sería un resultado razonablemente bueno (y en algunas fechas la tasa llega al 60 %) son únicamente los 15 activos con más probabilidad de pertenecer a la clase con más retorno los elegidos. Por ello se incide en que el resultado del backtest en la sección 5.3 es una representación más fiel del rendimiento de la estrategia.

Algoritmo	**Tasa de acierto media**	**Desviación de la tasa**
Decision Tree	0.292	0.05
Random Forest	0.293	0.051
XGBoost	0.404	0.104

Cuadro 5.1: Tabla de tasas de acierto

En la figura 5.5 podemos ver las tasas de acierto del XGBoost en cada fecha, y vemos como va manteniendo constantemente una mayor que los otros algoritmos aunque con el inconveniente de una desviación mayor.

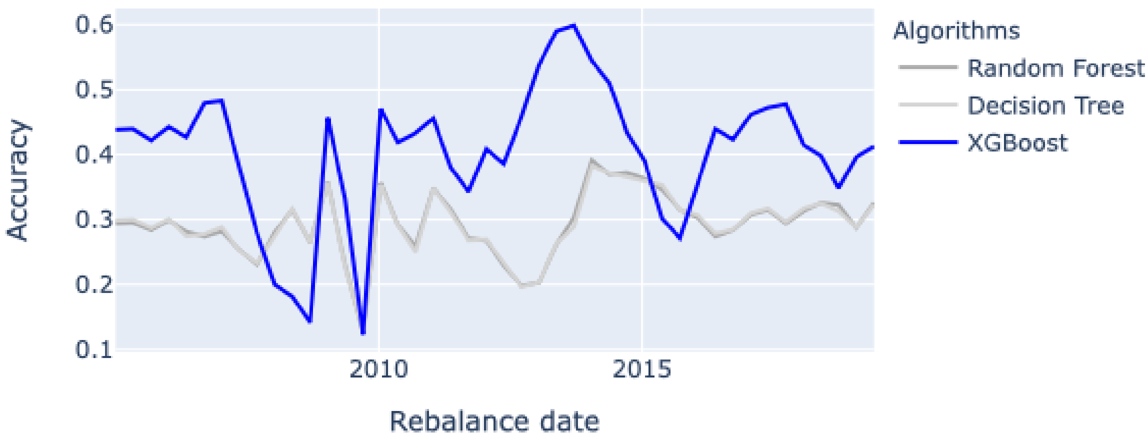

Figura 5.5: Tasas de acierto

5.2.1. Importancia de las características

La importancia de las características es diferente en el XGBoost respecto a los otros algoritmos, dado que xgb[4] es más complejo en este caso las importancias están más repartidas como podemos ver en la figura 5.6. Además de igualarse en importancia, las más

[4] *XGBoost*

relevantes cambian, y teniendo en cuenta las desviaciones de la figura hay diferentes clasificadores que cambian completamente de características más relevantes. Para los xgb no hay una predominancia clara de ninguna de las características. De igual forma que con el dt y el rf nos apoyaremos en la figura 5.7 que refleja las importancias a lo largo del tiempo de cada clasificador.

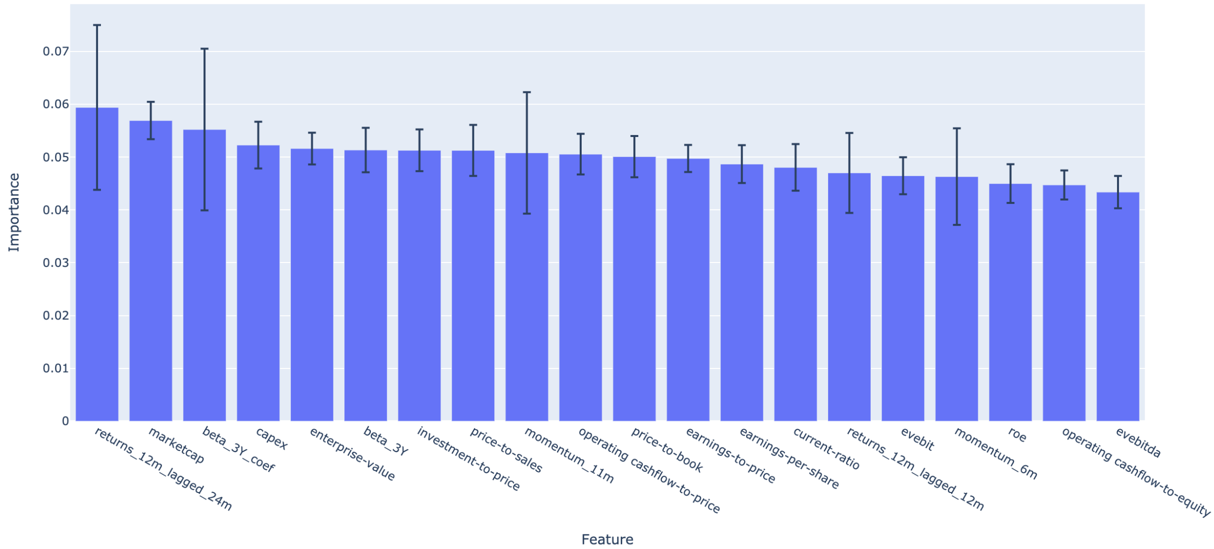

Figura 5.6: Ranking importancia de las características - XGBoost

En la figura 5.7 podemos observar:

- Nos encontramos un comportamiento parecido que en las figuras 5.3 y 5.4 en las que en diferentes fechas cambian las características más importantes. Sin embargo, las variaciones en las importancias son menores y se aprecia en el gráfico unos cambios más quirúrgicos para el xgb.

- También se puede observar como se da el mismo cambio drástico en importancias en el tramo de 2010 a 2015 que atribuimos a la crisis del 2008 y que se interpreta como un cambio en los factores que implicarán un mayor retorno en las compañías en función del contexto económico.

- Se produce adicionalmente que las características asociadas al valor (por ejemplo: evebit, evebitda, price-to-book y earnings-to-price) cobran importancia justo después de la crisis de 2008.

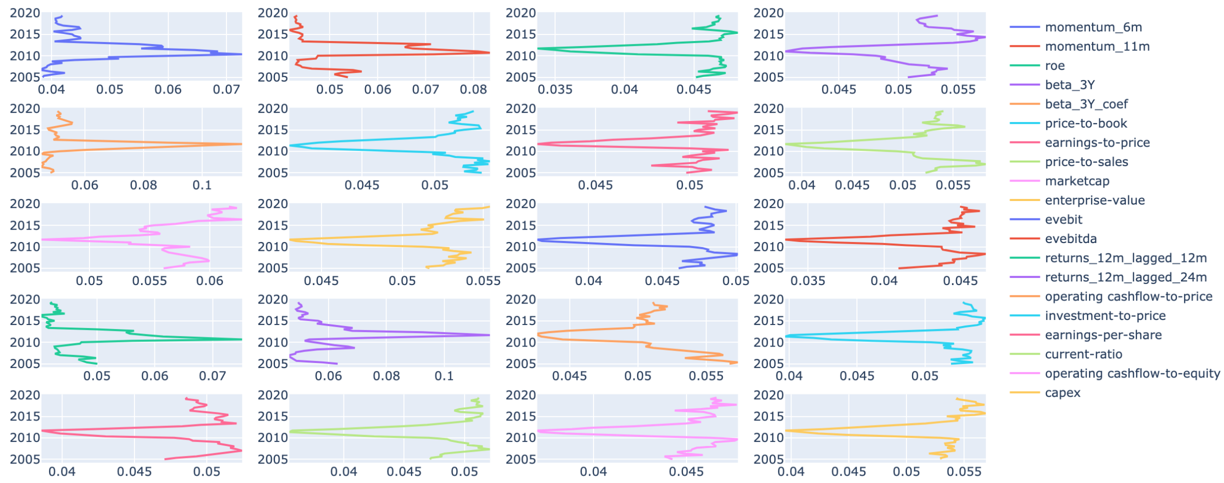

Figura 5.7: Importancia de las características - XGBoost

5.3. Resultado final

En esta sección mostramos los beneficios finales de las estrategias desarrolladas en comparación con el índice que se ha tomado de referencia (SP500). En la tabla 5.2 se encuentran los resultados de beneficios totales, anualizados y la volatilidad anual como medida de riesgo. Vemos como las estrategias han mejorado el rendimiento del mercado. Sin embargo, las estrategias basadas en el árbol de decisión y el bosque aleatorio tienen un riesgo muy elevado en comparación con el beneficio adicional que producen respecto al mercado. Al mismo tiempo la estrategia basada en el XGBoost consigue unos retornos mejores y aunque tiene más volatilidad que el SP500, ésta es muy similar a las de los otros algoritmos que no llegan a tener tan buen resultado en retornos.

Estrategia	**Retornos totales**	**Retornos anualizados**	**Volatilidad anual**
SP500	147.63 %	6.34 %	17.81 %
Strategy dt	189.63 %	7.48 %	33.54 %
Strategy rf	228.2 %	8.39 %	38.7 %
Strategy xgb	402.16 %	11.56 %	39.42 %

Cuadro 5.2: Tabla de retornos del backtest

En la figura 5.8 se muestra el resultado final de todas las estrategias así como el índice. Las 4 carteras comienzan con un valor de una cartera hipotética de 1$. No se han considerado en ningún caso los gastos y comisiones que se derivarían de la compra, venta y mantenimiento de los activos. Es importante recordar que en caso de llevar a la práctica la estrategia estos costes podrían suponer una disminución en la rentabilidad. Los activos mantenidos son 15 y los rebalanceos cada 4 meses, lo que implica que las transacciones podrían no ser muy numerosas y que los gastos derivados de comisiones no fueran elevados.

También en la figura se puede observar como las estrategias se comportan mejor que el índice cuando éste sube y peor que él cuando baja, sin embargo, las subidas acaban predominando sobre las bajadas, y a lo largo del tiempo las estrategias desbancan al índice.

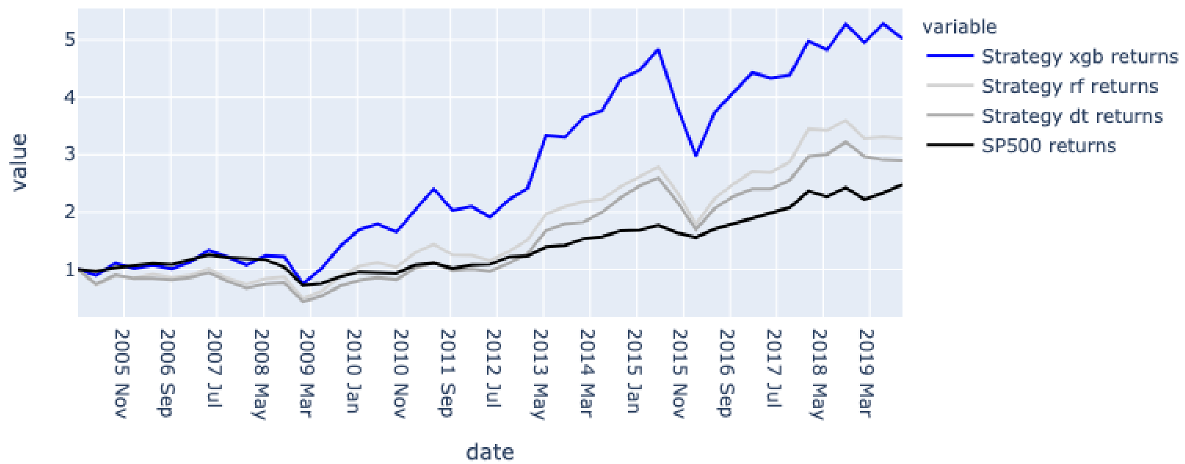

Figura 5.8: Retornos backtest

En resumen, en el presente trabajo se consigue un rendimiento del doble que el índice en el periodo estudiado de unos 14 años. Además, se ha sostenido a lo largo del tiempo a pesar de los movimientos más bruscos que padece la estrategia como se ve en la volatilidad anual.

Capítulo 6

Conclusiones y trabajo futuro

Los mercados financieros y la inversión es un tema de interés para académicos e inversores (tanto particulares como institucionales). Los modelos multifactor se llevan desarrollando décadas y recientemente se aplican algoritmos de aprendizaje automático para mejorar los modelos y capturar relaciones no lineales. Se están adoptando cada vez más técnicas de aprendizaje automático en finanzas, no solo en el campo de los mercados financieros, implicando desafíos para los reguladores y los gestores.

En el presente trabajo se han utilizado características (factores) utilizados por otros estudios obtenidos de un proveedor para garantizar la calidad de los datos. Los experimentos realizados se han centrado en mejorar los beneficios frente a una estrategia pasiva de inversión en el índice SP500 eligiendo un subconjunto de los activos que lo componen cada 4 meses.

Como resultado se ha comprobado que una estrategia utilizando factores presentes en la literatura y XGBoost se pueden conseguir retornos superiores al índice de referencia. En este caso un 5 % anual más que el índice a lo largo de 16 años. El beneficio, sin embargo, tiene como consecuencia una mayor volatilidad de la cartera y mayor riesgo.

Recientemente la inversión pasiva en índices tiene bastante éxito y generalmente es difícil de batir por la gestión activa (menos del 10 % de los gestores activos consiguen batir el mercado). Consideramos que este tipo de estrategias que pueden implementarse de manera pasiva, pueden proporcionar un mayor beneficio asumiendo más riesgo sin incurrir en los costes elevados de la gestión activa.

Se muestran a continuación potenciales líneas abiertas para investigación futura:

- Es posible ampliar o cambiar los factores utilizados para el entrenamiento. Podrían utilizarse otros factores presentes en la literatura o avanzar la investigación en la búsqueda de nuevos factores que expliquen el retorno de los activos.

- Realizar esta misma aproximación de seleccionar un subconjunto de activos pertenecientes a un índice pero aplicado en otros índices (por ejemplo: NASDAQ100, FTSE100, IBEX35).

- Ampliar el universo de activos contemplados a un mercado en su totalidad o varios mercados de geografías distintas (por ejemplo todos los activos de la bolsa en USA).

Al contemplarse un número mayor de activos los tiempos de cómputo y necesidades de capacidad de procesamiento serán mayores pero mejores oportunidades podrían encontrarse.

- Teniendo en cuenta que los factores importantes cambian en función del contexto económico una línea de investigación podría ser guardar los modelos generados en cada fecha y combinarlos con un modelo que pronostique el contexto económico general, de manera que aprovechemos los modelos previamente entrenados en situaciones análogas.

- Las técnicas utilizadas han sido derivadas de los árboles de decisión, utilizar redes neuronales con los hiperparámetros optimizados mediante algoritmos genéticos podría investigarse.

- En el mundo de la inversión tanto inversores como reguladores se preocupan por la explicabilidad de los modelos utilizados en este contexto. No solo los modelos deben producir buen resultado, además deben poder explicarse. Una línea futura de investigación podría ser, partiendo de modelos como el del presente trabajo, aplicar técnicas de explicabilidad del modelo para confirmar como funciona. En caso de que inversores o reguladores requieran explicaciones para el comportamiento del modelo éste será explicable a partir de las características del modelo de una forma clara.

Bibliografía

Arnott, R. D., Harvey, C. R., and Markowitz, H. (2018). A Backtesting Protocol in the Era of Machine Learning. *SSRN Electronic Journal*, pages 1–18.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Brogaard, J. and Zareei, A. (2018). Machine Learning and the Stock Market. *SSRN Electronic Journal*.

Caparrini López, A. and Pérez Molina, L. (2017). Clasificador de subgéneros de música electrónica.

Chen, T. and Guestrin, C. (2016). XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, New York, NY, USA. ACM.

Cochrane, J. H. (2011). Presidential Address: Discount Rates. *The Journal of Finance*, 66(4):1047–1108.

DeMiguel, V., Garlappi, L., and Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *Review of Financial Studies*, 22(5):1915–1953.

Fabozzi, F. J. and De Prado, M. L. (2018). Being honest in backtest reporting: A template for disclosing multiple tests. *Journal of Portfolio Management*, 45(1):141–147.

Fama, E. F. and French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22.

Fama, F. and French, R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33:3–56.

Feng, G., Giglio, S., and Xiu, D. (2020). Taming the Factor Zoo: A Test of New Factors. *Journal of Finance*, 75(3):1327–1370.

Fu, Y., Cao, S., and Pang, T. (2020). A sustainable quantitative stock selection strategy based on dynamic factor adjustment. *Sustainability (Switzerland)*, 12(10):1–12.

Harvey, C. R., Liu, Y., and Zhu, H. (2015). ... and the Cross-Section of Expected Returns. *Review of Financial Studies*, 29(1):5–68.

Hou, K., Xue, C., and Zhang, L. (2018). Replicating Anomalies. *The Review of Financial Studies*, 33(5):2019–2133.

Jidong, L. and Ran, Z. (2018). Dynamic Weighting Multi Factor Stock Selection Strategy Based on XGboost Machine Learning Algorithm. In *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, pages 868–872. IEEE.

Kakushadze, Z. and Yu, W. (2019). Machine Learning Risk Models. *SSRN Electronic Journal*.

Li, J. and Zhang, R. (2019). Dynamic Weighting Multi Factor Stock Selection Strategy Based on XGboost Machine Learning Algorithm. In *Proceedings of 2018 IEEE International Conference of Safety Produce Informatization, IICSPI 2018*, pages 868–872. IEEE.

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1):77.

Mclean, R. D. and Pontiff, J. (2016). Does Academic Research Destroy Stock Return Predictability? *Journal of Finance*, 71(1):5–32.

Patel, J., Shah, S., Thakkar, P., and Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications*, 42(4):2162–2172.

Rasekhschaffe, K. C. and Jones, R. C. (2019). Machine Learning for Stock Selection. *Financial Analysts Journal*, 75(3):70–88.

Sharpe, W. (1964). CAPITAL ASSET PRICES: A THEORY OF MARKET EQUILIBRIUM UNDER CONDITIONS OF RISK. *The Journal of Finance*, XIX:425–442.

Sugitomo, S. and Minami, S. (2018). Fundamental Factor Models Using Machine Learning. *Journal of Mathematical Finance*, 08(01):111–118.

Turunen, S. (2019). Feasibility of Nonlinear Multifactor Classifiers for Predicting Share Returns.