



MÁSTER EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

Nuevos modelos de asesoramiento automatizado de
inversiones ESG a través de inteligencia artificial

Autor: Patricia Samper Lario

Director: Alejandro González San Román

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título Nuevos modelos de asesoramiento automatizado de inversiones ESG a través de inteligencia artificial en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2021/2022 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Patricia Samper Lario Fecha: 11/07/2022



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Alejandro González San Román

Fecha: 11/07/2022



MÁSTER EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

Nuevos modelos de asesoramiento automatizado de inversiones ESG a través de inteligencia artificial.

Autor: Patricia Samper Lario

Director: Alejandro González San Román

Madrid

NUEVOS MODELOS DE ASESORAMIENTO AUTOMATIZADO DE INVERSIONES ESG A TRAVÉS DE INTELIGENCIA ARTIFICIAL

Autor: Samper Lario, Patricia.

Director: González San Román, Alejandro.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

Este Proyecto consta de **dos partes, una primera donde se hace un análisis** de los posibles usos y formas de incorporar la Inteligencia Artificial en los modelos de asesoramiento financiero conocidos como “Robo Advisors” y una **segunda parte práctica** que consiste en la creación del modelo de una aplicación Web a modo “Robo-Advisor” que incorpore alguno de los algoritmos explicados en la parte teórica.

Palabras clave: Sistemas de recomendación, Inversión de impacto, ETF, colocación de activos, Robo-Advisors, Inteligencia artificial.

1. Introducción

El documento comienza con una introducción a los temas de **inteligencia artificial, Robo-Advisors** e **inversión sostenible**.

Una vez hecho esto se entra a analizar los actuales Robo-advisor que existen y están liderando el mercado actualmente, para después proponer un modelo de asesoramiento automatizado que empleará alguna de **las técnicas de “machine learning”** vistas y además ofrecerá **recomendaciones personalizadas** de inversión.

Finalmente se procede a la creación de dicho modelo como un prototipo en formato de **aplicación Web**.

Una vez concluido todo el proceso se extraerán resultados, conclusiones y un análisis de futuras mejoras o tendencias.

En la siguiente imagen se presenta la estructura que seguirá el proyecto de una forma más visual.

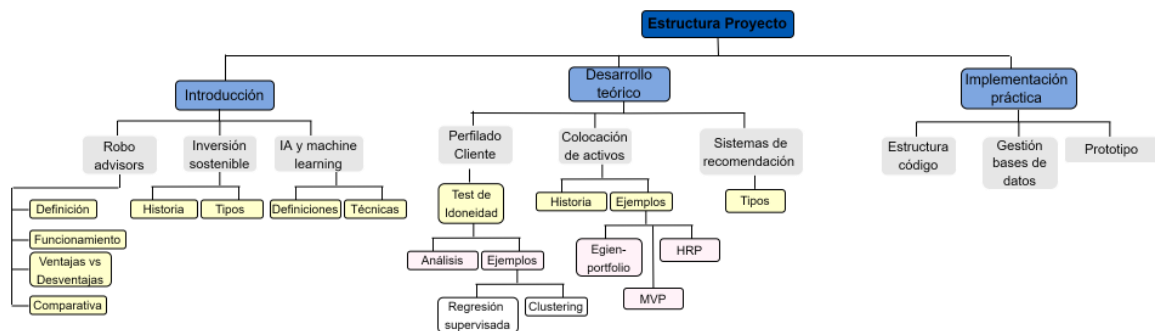


Figure 1-1: Esquema de la estructura del proyecto.

2. Definición del Proyecto

Este Proyecto pretende tratar algunos de los temas de moda que existen actualmente en el mundo financiero. Por un lado, se hablará de conceptos como Fintech, Inteligencia Artificial, inversión de impacto...

Todos estos temas están a la orden del día y está provocando importantes cambios en todo el sector de la gestión de activos.

Teniendo en cuenta que la evolución de los productos y servicios suele ir dirigida o bien hacia la **reducción de costes** o ahora más que nunca a **aumentar el grado de personalización** de estos, y los RAs una vez han conseguido el objetivo de reducir los costes gracias a la automatización de sus procesos se irán cada vez más enfocando en aumentar la adaptabilidad del producto a cada uno de sus usuarios. Es por ello por lo que todas las técnicas, ejemplos y métodos que se han nombrado a lo largo del proyecto van dirigidas a conseguir ese mayor grado de personalización que el mercado busca y valora cada vez más.

Se pretende ofrecer al lector una visión de la **situación actual** de estas nuevas **tendencias** y sus perspectivas de futuro.

Una vez establecido el contexto en el que nos encontramos, se procederá a una descripción más detallada a cerca de métodos y modelos concretos que emplean la tecnología de Inteligencia Artificial y Machine Learning, que pueden emplearse para la gestión de carteras de inversión, sistemas de recomendación y perfilado de clientes. De los cuales se escogerán aquellos que se perciban más adecuados para el objetivo de proyecto y se procederá a la construcción y desarrollo de una herramienta de asesoramiento digital (Robo-Advisor) en materia ESG. Ofreciendo al cliente la cartera constituida a partir de ETFs temáticos con los que el cliente se siente identificado y a gusto invirtiendo su dinero.

3. Descripción del modelo/sistema/herramienta

La implementación práctica de este proyecto consiste en la creación de una aplicación web a modo Robo-Advisor donde los usuarios puedan registrarse y configurar sus carteras de forma óptima y constituida por aquellos ETFs que más se ajusten a sus preferencias.

Para llevar a cabo este prototipo las herramientas de las que se han hecho uso son:

1. **Python:** Este es el lenguaje de programación principal que se ha empleado para la implementación de la aplicación.
2. **Stramlit:** Para implementar el Front-end de la web se ha hecho uso de la librería de Python conocida como streamlit.
3. **SQL:** Para la gestión de la base de datos donde se almacena toda la información que se va recogiendo de los diferentes usuarios se ha recurrido al lenguaje SQL.

4. **MySQL:** MySQL es un sistema de gestión de bases de datos relacional, donde se almacenará la base de datos.

4. Resultados

La tendencia del desarrollo de productos y servicios va cada vez más orientada hacia una mayor personalización de estos. Con este objetivo el proyecto se ha centrado en analizar diferentes técnicas que ayuden a estos sistemas de asesoramiento automatizado a aumentar su grado de personalización y flexibilidad.

Analizando dos de los procesos fundamentales de los Robo-Advisors actuales (perfilado de cliente y colocación de activos) y añadiendo un tercer proceso que viene a satisfacer la necesidad creciente por parte de las nuevas generaciones de seleccionar marcas y empresas que se alineen con sus valores.

Es por ello por lo que las conclusiones se han dividido en los siguientes apartados:

- En cuanto a la implementación de **la IA para el perfilado del cliente** considero que pueden ser muy interesantes para complementar el sistema actual, ya que muchas veces los propios inversores no estiman bien su capacidad de tolerar el riesgo. Aunque para ello es imprescindible contar con los **datos y la información necesaria** que permitan crear modelos fiables.
- En cuanto al uso de **la AI para la creación de carteras**, están surgiendo nuevas técnicas y modelos que pueden ayudar a establecer un **reparto de capital más eficaz y óptimo**, creando así portafolios más robustos. Muchas de ellas ya han demostrado un mejor rendimiento en comparación con metodologías anteriores.
- En cuanto al uso de sistemas de recomendación para productos ESG se ha llegado a la conclusión de que **sistemas sencillos** pueden ofrecer buenos resultados, y no es tanto el propio sistema como lo es la **obtención, clasificación y gestión de los datos e información** lo que determinará la calidad de las recomendaciones.



Figure 1-2: Ejemplo aplicación de recomendación (additiv Impact Investing with Clarity AI)

5. Referencias

- [1] Banik, R. (2018). *Hands-On Recommendation Systems with Python: Start building powerful and personalized, recommendation engines with Python* (English Edition) (1.a ed.). Packt Publishing.
- [2] Boreiko, D. (2020). *How Risk Profiles of Investors Affect Robo-Advised Portfolios*. *Frontiers*. Recuperado 24 de mayo de 2022, de <https://www.frontiersin.org/articles/10.3389/frai.2020.00060/full>
- [3] C. (2020, 4 diciembre). *Guía para entender los diferentes enfoques de inversión de la ESG y poder clasificar los fondos*. FundsPeople España. Recuperado 24 de mayo de 2022, de <https://fundspeople.com/es/guia-para-entender-los-diferentes-enfoques-de-inversion-de-la-esg-y-poder-clasificar-los-fondos/>
- [4] Finanbest. (2022, 18 mayo). *Home*. Gestor automatizado de inversión | Líder en rentabilidad. Recuperado 24 de mayo de 2022, de <https://www.finanbest.com/>
- [5] *Goals Archive*. (2022, 15 marzo). The Global Goals. Recuperado 24 de mayo de 2022, de <https://www.globalgoals.org/goals/>
- [6] Kane, F. (2021). *Building Recommender Systems with Machine Learning and AI: Help people discover new products and content with deep learning, neural networks, and machine learning recommendations*. Independently published.
- [7] Lewinson, E. (2022). *Python for Finance Cookbook: Over 60 powerful recipes for effective financial data analysis, 2nd Edition*. Packt Publishing - ebooks Account.
- [8] N. (2021, 13 octubre). *Turning client suitability into a science*. Neuroprofiler. Recuperado 24 de mayo de 2022, de <https://neuroprofiler.com/en/client-suitability/>

- [9] PricewaterhouseCoopers. (2021). *AI for asset and wealth managers*. PwC. Recuperado 24 de mayo de 2022, de <https://www.pwc.com/us/en/tech-effect/ai-analytics/ai-predictions/asset-and-wealth-management.html>
- [10] S. (2022, 24 mayo). *How to Develop a FinTech App in 2022 A Guide to Action*. SpdLoad. Recuperado 7 de junio de 2022, de <https://spdload.com/blog/fintech-app-development/>
- [11] Sironi, P. (2016). *Fintech Innovation: From Robo-Advisors to Goal Based Investing and Gamification* (1. ed.). Wiley.
- [12] Tatsat, H., Puri, S., & Lookabaugh, B. (2020). *Machine Learning and Data Science Blueprints for Finance: From Building Trading Strategies to Robo-Advisors Using Python*. O'Reilly Media.
- [13] M. Day, T. Cheng and J. Li, "AI Robo-Advisor with Big Data Analytics for Financial Services," 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2018, pp. 1027-1031, doi: 10.1109/ASONAM.2018.8508854.
- [14] Zibriczky12, D. (2016). Recommender systems meet finance: a literature review. In Proc. 2nd Int. Workshop Personalization Recommender Syst (pp. 1-10).

NEW MODELS OF AUTOMATED ESG INVESTMENT ADVICE THROUGH ARTIFICIAL INTELLIGENCE

Author: Samper Lario, Patricia.

Supervisor: González San Román, Alejandro.

Collaborating Entity: ICAI – Universidad Pontificia Comillas.

ABSTRACT

This project consists of **two parts**, a **first part** where an **analysis** of the possible uses and ways of incorporating Artificial Intelligence in the financial advisory models known as "Robo Advisors" is made, and a **second practical part** that consists of the creation of a model of a **Web application** as a "Rob-Advisor" that incorporates some of the algorithms explained in the theoretical part.

Keywords: Recommender systems, Impact investing, ETFs, asset allocation, Robo-Advisors, Artificial Intelligence.s

1. Introduction

The document begins with an introduction to the topics of artificial intelligence, Robo-Advisors and sustainable investment.

Once this is done, the paper analyzes the current Robo-advisors that exist and are currently leading the market, and then proposes an automated advice model that will use some of the "machine learning" techniques seen and will also offer personalized investment recommendations. Finally, the model is created as a prototype in the form of a Web application.

Once the whole process has been completed, results, conclusions and an analysis of future improvements or trends will be drawn.

The following image shows the structure that the project will follow in a more visual way.

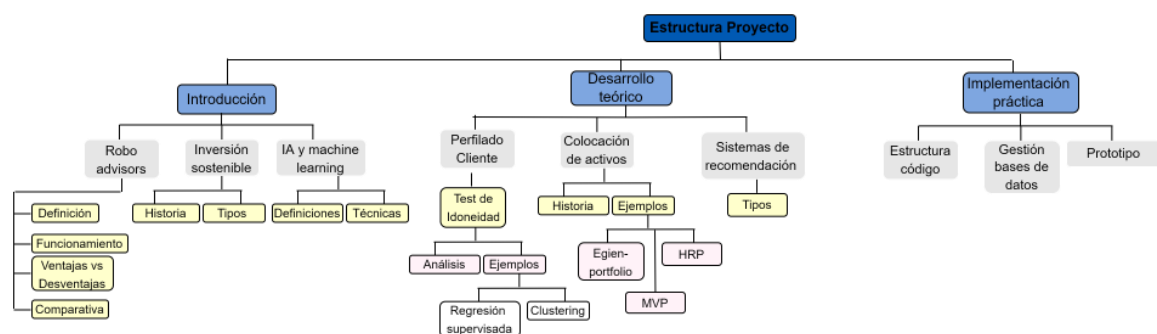


Figure 1-3: Outline of the project structure.

2. Project Description

This project aims to address some of the current fashionable topics in the financial world. On the one hand, we will talk about concepts such as **Fintech, Artificial Intelligence, impact investing ...**

All these topics are the order of the day and are causing important changes in the entire asset management sector. It is intended to provide the reader with an overview of the **current status** of these new trends and their **future prospects**.

Considering that the evolution of products and services is usually directed either towards **cost reduction** or now more than ever to **increase the degree of customization** of these, so the RAs once they have achieved the goal of reducing costs by the automation of their processes will be more and more focus on **increasing the adaptability** of the product to each of its users. That is why all the techniques, examples and methods that have been mentioned throughout the project are aimed at achieving the greater degree of customization that the market seeks.

Once the context in which we find ourselves has been established, we will proceed to a more detailed description of specific methods and models using Artificial Intelligence and Machine Learning technology, which can be used for investment portfolio management, recommendation systems and client profiling.

From which we will choose those that are perceived to be most suitable for the project objective and proceed to the construction and development of a digital advisory tool (Robo-Advisor) in ESG matters. Offering the client, the portfolio constituted of thematic ETFs with which the client feels identified and comfortable investing his money.

3. Model/system/tool description

The practical implementation of this project consists of the creation of a web application as a Robo-Advisor where users can register and configure their portfolios in an optimal way and made up of those ETFs that best suit their preferences.

To carry out this prototype, the tools that have been used are:

1. Python: This is the main programming language used for the implementation of the application.

2. Stramlit: To implement the Front-end of the web we have made use of the Python library known as streamlit.

3. SQL: For the management of the database where all the information collected from the different users is stored, the SQL language has been used.

4. MySQL: MySQL is a relational database management system, where the database will be stored.

4. Results and conclusions

The trend in the development of products and services is increasingly oriented towards greater personalization. With this objective in mind, the project has focused on analyzing different techniques that help these automated advisory systems to increase their degree of personalization and flexibility.

Analyzing two of the fundamental processes of current Robo-Advisors (customer profiling and asset placement) and adding a third process that comes to meet the growing need of new generations to select brands and companies that align with their values.

That is why the conclusions have been divided into the following sections:

- Regarding the implementation of **AI for client profiling** I consider that they can be very interesting to complement the current system, since many times investors themselves do not estimate well their capacity to tolerate risk. But it will be essential to have the **necessary data and information** to create reliable models.
- As for the use of **AI for portfolio creation**, new techniques and models are emerging that can help establish a **more efficient and optimal allocation of capital**, thus creating more robust portfolios. Many of these have already demonstrated improved performance compared to previous methodologies.
- Regarding the use of **recommendation systems for ESG products**, it has been concluded that **simple systems** can provide good results, and it is not so much the system itself as the **collection, classification and management of data and information** that will determine the quality of the recommendations.

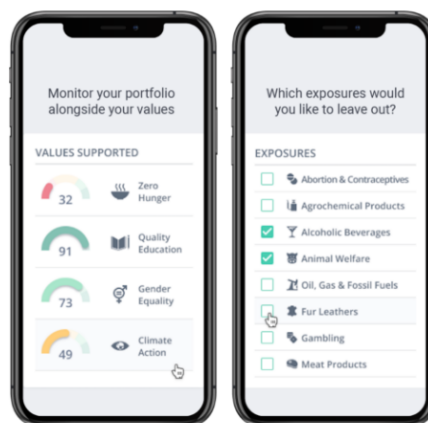


Figure 1-4: Example of a recommendation app (additiv Impact Investing with Clarity AI)

5. References

- [15] Banik, R. (2018). Hands-On Recommendation Systems with Python: Start building powerful and personalized, recommendation engines with Python (English Edition) (1.a ed.). Packt Publishing.

- [16] Boreiko, D. (2020). *How Risk Profiles of Investors Affect Robo-Advised Portfolios*. Frontiers. Recuperado 24 de mayo de 2022, de <https://www.frontiersin.org/articles/10.3389/frai.2020.00060/full>
- [17] C. (2020, 4 diciembre). *Guía para entender los diferentes enfoques de inversión de la ESG y poder clasificar los fondos*. FundsPeople España. Recuperado 24 de mayo de 2022, de <https://fundspeople.com/es/guia-para-entender-los-diferentes-enfoques-de-inversion-de-la-esg-y-poder-clasificar-los-fondos/>
- [18] Finanbest. (2022, 18 mayo). *Home*. Gestor automatizado de inversión | Líder en rentabilidad. Recuperado 24 de mayo de 2022, de <https://www.finanbest.com/>
- [19] *Goals Archive*. (2022, 15 marzo). The Global Goals. Recuperado 24 de mayo de 2022, de <https://www.globalgoals.org/goals/>
- [20] Kane, F. (2021). *Building Recommender Systems with Machine Learning and AI: Help people discover new products and content with deep learning, neural networks, and machine learning recommendations*. Independently published.
- [21] Lewinson, E. (2022). *Python for Finance Cookbook: Over 60 powerful recipes for effective financial data analysis, 2nd Edition*. Packt Publishing - ebooks Account.
- [22] N. (2021, 13 octubre). *Turning client suitability into a science*. NeuroProfiler. Recuperado 24 de mayo de 2022, de <https://neuroprofiler.com/en/client-suitability/>
- [23] PricewaterhouseCoopers. (2021). *AI for asset and wealth managers*. PwC. Recuperado 24 de mayo de 2022, de <https://www.pwc.com/us/en/tech-effect/ai-analytics/ai-predictions/asset-and-wealth-management.html>
- [24] S. (2022, 24 mayo). *How to Develop a FinTech App in 2022 A Guide to Action*. SpdLoad. Recuperado 7 de junio de 2022, de <https://spdload.com/blog/fintech-app-development/>
- [25] Sironi, P. (2016). *Fintech Innovation: From Robo-Advisors to Goal Based Investing and Gamification* (1. ed.). Wiley.
- [26] Tatsat, H., Puri, S., & Lookabaugh, B. (2020). *Machine Learning and Data Science Blueprints for Finance: From Building Trading Strategies to Robo-Advisors Using Python*. O'Reilly Media.
- [27] M. Day, T. Cheng and J. Li, "AI Robo-Advisor with Big Data Analytics for Financial Services," 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2018, pp. 1027-1031, doi: 10.1109/ASONAM.2018.8508854.
- [28] Zibriczky12, D. (2016). Recommender systems meet finance: a literature review. In Proc. 2nd Int. Workshop Personalization Recommender Syst (pp. 1-10).

Índice de la memoria

Capítulo 1. Introducción	10
1.1 Motivación del proyecto	10
Capítulo 2. Descripción de las Tecnologías.....	12
Capítulo 3. Estado de la Cuestión.....	13
3.1 Robo-advisors	13
3.1.1 Definición.....	13
3.1.2 ¿Cómo funcionan?.....	13
3.1.3 Ventajas y desventajas.....	15
3.1.4 Ejemplos y comparativa	16
3.2 Inversión sostenible.....	16
3.2.1 RSI : Inversión socialmente responsable.....	18
3.3 Machine Learning , Deep Learning, Artificial Intelligence and Data Science.....	19
3.3.1 Machine learning.....	21
Capítulo 4. Desarrollo Teórico.....	26
4.1 Introducción.....	26
4.2 Perfilado del cliente.....	27
4.2.1 Test de idoneidad (MIFID)	27
4.3 Colocación de activos	36
4.3.1 Historia de la optimización de carteras	36
4.3.2 Markowitz model: Mean-Variance Optimization	37
4.3.3 Eigen -Portfolio	39
4.3.4 Hierarchical Risk Parity.....	41
4.4 Sistemas de recomendación.....	43
4.4.1 El problema de recomendación.....	43
.....	43
4.4.2 Tipos de sistemas de recomendación	44
4.4.3 Algunas métricas de evaluación.....	55
4.4.4 Datos	58
4.4.5 Modelos de recomendación empleados en el prototipo	59

Capítulo 5. Implementación Práctica: Modelo Robo-Advisor.....	60
5.1 Diseño Interfaz.....	60
5.2 Código Base.....	62
5.3 Código SQL.....	65
5.4 Código de Recomendaciones.....	70
5.4.1 Top 5 Seleccionados.....	70
5.4.2 Sistema de recomendación basado en la descripción.....	71
5.4.3 Sistema basado en contenido a partir de las características de los ETFs	72
5.4.4 Filtro colaborativo simple basado en el usuario.....	73
5.4.5 SVD.....	74
5.4.6 Sistema Híbrido 1	75
5.4.7 Sistema Híbrido 2	77
5.5 Código colocación de activos	78
5.6 Código Cálculo Jubilación	90
Capítulo 6. Estudio económico	<i>Error! Bookmark not defined.</i>
6.1 Pasos para el desarrollo.....	95
6.2 Proyección financiera	97
6.2.1 Proyección Costes.....	97
6.2.2 Proyección Ingresos.....	99
Capítulo 7. Análisis de resultados, conclusiones y trabajos futuros	101
Capítulo 8. Alineación con los Objetivos de Desarrollo Sostenible.....	111
Capítulo 9. Bibliografía.....	112
ANEXO I	114

Índice de Figuras

Figure 1-1:Esquema de la estructura del proyecto.	5
Figure 1-2:Ejemplo aplicación de recomendación (additiv Impact Investing with Clarity AI)	8
Figure 1-3:Outline of the project structure.	10
Figure 1-4:Example of a recommendation app (additiv Impact Investing with Clarity AI)	12
Figure 3-3-1:Flujos de fondos sostenibles.Morningstar (March 2021).....	17
Figure 3-3-2:BlackRock estimacion de activos de Fondos y ETFs sostenibles	17
Figure 3-3:AI, Machine learning,deep learningand data science.(libro “Machine Learning and Data Science for Finance”.).....	20
Figure 3-4: https://www.globalbankingandfinance.com/artificial-intelligence-ai-in-bfsi-market-to-grow-at-30-cagr-from-2018-2024/	21
Figure 3-5:Tipos dentro de MACHine learning(https://www.freegyann.com/post/build-a-model-with-python).....	22
Figure 4-1:Variables predictivas(Libro “Machine Learning and Data Science Blueprints for Finance”)	33
Figure 4-2:Influencia de las variables en el modelo (Libro “Machine Learning and Data Science Blueprints for Finance”).....	33
Figure 4-3: Resultado final del modelo (Libro “Machine Learning and Data Science Blueprints for Finance”).....	34
Figure 4-4:Atributos para la agrupación de individuos. (libro “Machine Learning and Data Science Blueprints for Finance”).....	35
Figure 4-5:Conclusiones algoritmo.(Libro: "Machine Learning and Data Science Blueprints for Finance").....	35
Figure 4-6: Efficient Frontier (Wikipedia).....	37
Figure 4-7:Efficient Frontier y curvas de indiferencia (Wikipedia).....	38

Figure 4-8:Diferentes Eigen Portfolios (libro “Machine Learning and Data Science Blueprints for Finance”).....	40
Figure 4-9: Resultados de los diferentes porfolios.	40
Figure 4-10:Distribución de activos Portfolio 0 (libro “Machine Learning and Data Science Blueprints for Finance”).....	41
Figure 4-11:Dendrogram (Libro “Machine Learning and Data Science Blueprints for Finance”).....	41
Figure 4-12:Quiasi-diagonalization (libro “Machine Learning and Data Science Blueprints for Finance”).....	42
Figure 4-13:: Colocación activos MVP vs HRP(libro “Machine Learning and Data Science Blueprints for Finance”)......	42
Figure 4-14:Esquema problema de recomendación.	43
Figure 4-15:Aplicación PCA a la Matriz usuario-producto.	52
Figure 4-16:Aplicación PCA a la transpuesta de la matriz usuario producto.....	53
Figure 4-17: PCA (Libro:Hands on recomendation systems with Python).....	53
Figure 4-18:Representación de las puntuaciones como producto matricial.	54
Figure 4-19: Esquema sistemas de recomendación.....	57
Figure 5-1: Esquema de la estructura de la aplicación Web.....	60
Figure 5-2:Estructura base de datos (Tablas)	65
Figure 5-3:Ejemplo del gráfico de dispersión de un usuario para 1000 portafolios diferentes	80
Figure 5-4:Ejemplo resultado de optimización MVP para la cartera de un usuario.....	83
Figure 5-5:Ejemplo optimización cartera usuario.	85
Figure 5-6:Ejemplo portafolios MVP y HRP usuario.	89
Figure 5-7:Ejemplo Backtest MVP vs HRP	89
Figure 5-8:Comparación resultados del desempeño “in sample” vs “out of sample”	90
Figure 5-9:Ejemplo Cálculo 1(Retirement Simulation)	92
Figure 5-10:Ejemplo Cálculo 2 (Retiremt Simulation).....	94
Figure 6-1: Gráfico evolución ingresos por escenarios.....	100

Figure 7-1: Como AI ayuda a los gestores de carteras a mejorar la experiencia de los usuarios (AI for asset and wealth managers in 2021: PwC)	103
Figure 7-2: additiv Impact Investing with Clarity AI.....	108

Índice de tablas

Tabla 3-1:Listado de Fondos Finanbest(https://www.finanbest.com/)	15
Tabla 3-2: Ventajas vs Desventajas (Robo-advisiros).....	15
Tabla 3-3:Comparativa Rovo-Advisors (https://www.roboadvisors.es/).....	16
Tabla 3-4:Tipos de inversión socialmente responsable.....	19
Tabla 4-1:Tabla características de los ETFs.....	58
Tabla 6-1: Predicción costes iniciales	98
Tabla 6-2: Proyección costes estimados.....	98
Tabla 6-3: Proyección según los diferentes escenarios	99

Índice de Ecuaciones

Ecuación 4-1: Ecuación Nota ponderada.....	45
Ecuación 4-2: Peso asignado a cada palabra.	46
Ecuación 4-3: Cálculo similitud coseno (Cosine similarity).....	47
Ecuación 4-4:Cálculo similitud coseno ajustada	47
Ecuación 4-5:Similitud de Pearson.....	47
Ecuación 4-6: Grado de similitud entre usuarios (Hands-On Recommendation Systems with Python).....	50
Ecuación 4-7: Similitud entre usuarios normalizada (Building Recommender Systems ??)	50
Ecuación 4-8:Ecuación para SVD.	54
Ecuación 4-9:Ecuación Accuracy.....	55
Ecuación 4-10: Ecuación del RMSE	55
Ecuación 4-11: Ecuación Precision.....	56
Ecuación 4-12: Ecuación Recall.....	56
Ecuación 4-13:Ecuación F1 Score.....	56

Índice de Código

Código 5-1: Función principal (main).....	62
Código 5-2: Opciones del main.....	63
Código 5-3: Funciones para la encriptación de contraseñas.....	63
Código 5-4.....	63
Código 5-5.....	64
Código 5-6:Funciones para la creación y manipulación de tablas(userstable).....	66
Código 5-7: Establecimiento de la conexión con la base de datos.....	66
Código 5-8: Funciones para la creación y manipulación de tablas(userspuntuation).	66
Código 5-9: Funciones para la creación y manipulación de tablas(usersselection).	67
Código 5-10: Funciones para transformar tabla SQL a DataFrame	67
Código 5-11	68
Código 5-12:Función para obtener respuestas del test de idoneidad del usuario.....	68
Código 5-13:Primera parte de la función que obtiene las puntuaciones de los ETFs por parte del usuario.....	69
Código 5-14:Primera parte de la función que obtiene las puntuaciones de los ETFs por parte del usuario.....	69
Código 5-15:Función que recoge la selección final del usuario.....	70
Código 5-16:Función Top 5 ETFs seleccionados.....	70
Código 5-17:Código del Sistema de recomendación basado en la descripción.	71
Código 5-18:Código del Sistema de recomendación basado en las características.....	72
Código 5-19: Preparación de los datos del modelo.....	73
Código 5-20:Filtro colaborativo según la similitud con otros usuarios.....	73
Código 5-21:Funciones para evaluar eficacia del modelo.....	74
Código 5-22: Filtro colaborativo SVD.	74
Código 5-23: Función que permite extraer las características de un ETF.....	75
Código 5-24: Función que despliega las recomendaciones.....	75
Código 5-25: Sistema Híbrido 1	76
Código 5-26: Sistema Híbrido 2.....	77

Código 5-27: Función iterative que crea diferentes portafolios asignados pesos aleatorios. .	78
Código 5-28	79
Código 5-29: Extracción del portafolio óptimo dentro de los anteriores.	79
Código 5-30.	81
Código 5-31: Obtención del portafolio óptimo mediante un problema de optimización.	82
Código 5-32: Evolución histórica de una cartera con los pesos obtenidos para el mejor portafolio.	82
Código 5-33: Cálculo del portafolio óptimo con MVP empleando librería ‘pypfopt’	84
Código 5-34: Cálculo del portafolio óptimo con HRP empleando librería ‘pypfopt’	84
Código 5-35	86
Código 5-36	86
Código 5-37	87
Código 5-38	87
Código 5-39	88
Código 5-40	88
Código 5-41: Función para el cálculo de la evolución del dinero.	91
Código 5-42: Presentación gráfica de resultados.	92
Código 5-43	93
Código 5-44: Cálculo jubilación 2.....	93

Capítulo 1. INTRODUCCIÓN

Este Capítulo pretende ofrecer una visión general de temas como las Fintech, Inteligencia Artificial, inversión de impacto que existen actualmente en el mundo financiero.

Todos estos temas están a la orden del día y está provocando importantes cambios en todo el sector de la gestión de activos.

Se pretende ofrecer al lector una visión de la situación actual de estas nuevas tendencias y sus perspectivas de futuro.

Una vez establecido el contexto en el que nos encontramos, se procederá a una descripción más detallada a cerca de métodos y modelos concretos que emplean la tecnología de Inteligencia Artificial y Machine Learning , que pueden emplearse para la gestión de carteras de inversión, sistemas de recomendación y perfilado de clientes. De los cuales se escogerán aquellos que se perciban más adecuados para el objetivo de proyecto y se procederá a la construcción y desarrollo de una herramienta de asesoramiento digital (Robo-Advisor) en materia ESG. Ofreciendo al cliente no solo opciones de inversión diversificadas con las que el cliente se siente identificado en cuanto a valores y principios se refieren.

1.1 MOTIVACIÓN DEL PROYECTO

El objetivo de este trabajo de fin de Máster es ofrece una visión global de las nuevas **tendencias** que se están dando dentro del mundo de las finanzas tanto el auge de la inversión sostenible y responsable como el creciente uso de nuevas tecnologías, como pueden ser el **asesoramiento automatizado** y empleo de técnicas de **inteligencia artificial**.

Se hablará en detalle de los conocidos como Robo-advisors (RA) y se diseñará un **prototipo** de que emplee técnicas de inteligencia artificial para el asesoramiento personalizado a los clientes aportando así mayor valor a los usuarios y ayudando e impulsando el interés por la inversión socialmente responsable.

Como se ha mencionado en puntos anteriores tanto la Inteligencia artificial, la inversión sostenible están en auge y todavía les queda un gran recorrido. Esto unido al cambio que se está produciendo a nivel mundial de la democratización de las finanzas brinda una excelente oportunidad para la unión de estas y la innovación en la creación de modelos más eficientes, inteligentes, y sostenibles.

Es por todo el potencial y la infinidad de posibilidades que ofrecen estas tendencias la razón por la cual se ha elegido este trabajo de fin de Máster.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

Aparte de los libros y artículos que se emplearán para la recolección de información se utilizarán otras herramientas que permitan la implantación de nuestro prototipo.

La implementación práctica de este proyecto consiste en la creación de una **aplicación web** a modo Robo-advisor donde los usuarios puedan registrarse y configurar sus carteras de forma óptima y constituida por aquellos ETFs que más se ajusten a sus preferencias.

Para llevar a cabo este prototipo las herramientas de las que se han hecho uso son:

1. **Python:** Este es el lenguaje de programación principal que se ha empleado para la implementación de la aplicación.
2. **Streamlit:** Para implementar el *Front-end* de la web se ha hecho uso de la librería de Python conocida como streamlit.
3. **SQL:** Para la gestión de la base de datos donde se almacena toda la información que se va recogiendo de los diferentes usuarios se ha recurrido al lenguaje SQL.
4. **MySQL:** MySQL es un sistema de gestión de bases de datos relacional, donde se almacenará la base de datos.
5. **mysql.connector :** Para la interacción entre la base de datos y el lenguaje de Python se importa mysql.connector que nos permitirá acceder a esta base de datos de nuestro código.
6. **ETF DataBase:** LA información de los diferentes ETFs que serán ofrecidos por nuestro Robo-Advisor se obtendrá de la página web, ETFDatabase.

Capítulo 3. ESTADO DE LA CUESTIÓN

Este capítulo ofrece un repaso acerca de las siguientes cuestiones: **Robo-advisors**, **Inversión sostenible** y socialmente responsable e **Inteligencia artificial**.

3.1 ROBO-ADVISORS

3.1.1 DEFINICIÓN

Podríamos entender los Robo-Advisors (RA) como un tipo de asesor financiero que ofrece un **servicio de gestión online** de carteras de inversión mediante **algoritmos**, **automatización** y, normalmente, la supervisión de un equipo de expertos inversores.

Los RA hacen posible crear carteras de **inversiones personalizadas** para cada persona y **adaptadas a sus posibilidades**.

3.1.2 ¿CÓMO FUNCIONAN?

Con el objetivo de ofrecer un servicio personalizado, como se ha visto anteriormente, el cliente en primer lugar realiza un **cuestionario** donde responde a preguntas relacionadas con la **cantidad** de dinero que desean invertir, el **rendimiento** que esperan obtener, y su tolerancia al **riesgo**. Suele consistir en una prueba de unas pocas preguntas (Test de idoneidad) que no lleve más de 5 minutos completar, cuyo objetivo es definir el tipo de inversor que eres según tu aversión al riesgo y las rentabilidades que buscas.

Analizaremos a modo de ejemplo el funcionamiento de los *Robo-advisor* “Finanbest”.

Finanbest:

1. Test de Idoneidad

En primer lugar, se realiza un **análisis individualizado**, esto es la prueba mencionada anteriormente. Como se indica en la web de Finanbest este consiste en una triple evaluación (**conocimiento y experiencia, objetivos y situación financiera**), a través del cual se obtiene el perfil del inversor según su capacidad y tolerancia al riesgo. En concreto Finanbest cuenta con 6 tipos de inversores que se traducen en **6 carteras modelo** diferentes para cada uno de estos perfiles.

2. Distribución de activos

El objetivo es la obtención de una distribución de activos que **optimice el binomio rentabilidad/riesgo**. Esta cartera estará formada por diferentes fondos de Inversión.

Es importante también tener en cuenta la **correlación** entre los diferentes activos que formen la cartera escogiendo aquellos poco correlacionados para optimizar la rentabilidad y minimizar el riesgo.

Los principios que emplea Finanbest son los de la **Teoría Moderna de Carteras (MPT)** en cuyos fundamentos se indagará más adelante.

Además de este modelo también aplican un esquema evolucionado de optimización basado en el Modelo de Black Litterman y complementan ambos con un comité de inversiones.

3. Selección de Fondos

En este caso los fondos con los que cuenta Finanbest y a partir de los cuales se crean las diferentes carteras son los siguientes:

Deuda Pública EUR Corto Plazo	Pictet	LU0167154417	0,30%	Activa
Deuda Pública EUR Largo Plazo	Vanguard	IE0007472990	0,12%	Indexada
Deuda Pública USA USD Corto Plazo	Eurizon	LU0338989397	0,24%	Indexada
Deuda Pública USA USD Largo Plazo	Pictet	LU1654546347	0,39%	Activa
Renta Fija Corp IG EUR Corto Plazo	Erste	AT0000A1XLV3	0,19%	Activa
Renta Fija Corp IG EUR Largo Plazo	BlackRock	IE00B67T5GZ1	0,15%	Indexada
Renta Fija Bonos ligados a Inflación EUR	BlackRock	IE00B4WXT857	0,13%	Indexada
Renta Fija Países Emergente H	M&G	LU1670632170	0,80%	Activa
Renta Fija High Yield Global H	Pimco	IE00B2R34Y72	0,55%	Activa
Renta Fija China Renminbi	Aberdeen	LU1834169366	0,60%	Activa
Renta Variable Países Desarrollados	Vanguard	IE00B03HD191	0,18%	Indexada
Renta Variable Países Emergentes	Amundi	LU0996175948	0,20%	Indexada
Renta Variable Small Caps	Vanguard	IE00B42W4L06	0,29%	Indexada
Renta Variable USA	Vanguard	IE0032126645	0,10%	Indexada
Renta Variable Europa	Amundi	LU0389811539	0,15%	Indexada
Renta Variable Japón	BlackRock	IE00B6RVVW34	0,17%	Indexada
Renta Variable Inmobiliario Indirecto	BlackRock	IE00B83YJG36	0,22%	Indexada

Tabla 3-1: Listado de Fondos Finanbest (<https://www.finanbest.com/>)

4. Reajuste

Con el fin de asegurar que se mantienen las proporciones de cada activo y así mantener el perfil de riesgo dentro de los parámetros establecidos se realiza un **reajuste automático** cuando alguno de los tipos de activo experimente una variación superior al límite establecido.

3.1.3 VENTAJAS Y DESVENTAJAS

En cuanto a las ventajas y desventajas que tienen los Robo-advisors frente a otros métodos de inversión se muestran en la siguiente tabla:

Ventajas	Desventajas
Bajo coste	No hay contacto con un asesor financiero real
Cierto grado de personalización	No es 100% personalizado: carteras modelo predefinidas
Mayor accesibilidad para los clientes	Poca profundidad en la determinación del perfil de riesgo
Mejor trazabilidad	
Asesoramiento eficiente y diversificado	
Simplicidad y facilidad para el usuario	
Evitan sesgos emocionales en las decisiones de inversión	

Tabla 3-2: Ventajas vs Desventajas (Robo-advisiros)

3.1.4 EJEMPLOS Y COMPARATIVA

A modo de ejemplo en la tabla inferior se ofrece una comparativa entre los principales Robo-Advisors que existen actualmente en España:


Robo Advisor					
Plan	Cartera fondos de inversión	Plan de inversión	Cartera de fondos indexados	inbestMe Strategic FI	Cartera de fondos
Tipo de contrato	Contrato gestión	Contrato gestión	Contrato gestión	Contrato gestión	Contrato gestión
Tipo de Activos	Renta variable, Renta fija	Renta variable y Renta fija y REITs	Renta variable y Renta fija	Renta variable, Renta fija y REITs	Renta variable y Renta fija
Nº de Carteras	10	5	5	11	9
Fondos incluidos	De 2 a 9	13	De 7 a 9	De 2 a 13	De 5 a 9
Ingreso min.	3.000 €	1.000 €	150€ (Aportación obligatoria)	1.000 €	3.000 €
Entidad custodia	Inversis o Cecabank	Inversis	Inversis	BNP Paribas	Cecabank
Comisión con aportación más baja	0,62%	0,65%	0,45%	0,74%	0,83%
Características	<ul style="list-style-type: none"> *Líder del mercado con más de 39.000 clientes *El mayor capital bajo gestión. 	<ul style="list-style-type: none"> *Comisiones más bajas y decrecientes. *Máxima diversificación *Incluye activos inmobiliarios globales *Alta totalmente online 	<ul style="list-style-type: none"> *Comisiones muy bajas *Es obligatorio hacer una aportación mensual mínima de 150€ 	<ul style="list-style-type: none"> *Combinación de gestión activa y pasiva *Alta online *Dispone de carteras temáticas 	<ul style="list-style-type: none"> *Gestión activa y pasiva *Carteras diversificadas

Tabla 3-3: Comparativa Robo-Advisors (<https://www.roboadvisors.es/>)

Como se puede ver, el número de carteras que ofrecen estos Robo-Advisors oscila entre 5 hasta 10 y los fondos que pueden incluir no superan los 13.

3.2 INVERSIÓN SOSTENIBLE

Es indiscutible que en los últimos años el interés general por el desarrollo sostenible y una mayor concienciación por parte de los ciudadanos se ha visto reflejado también en el mundo de las inversiones surgiendo así lo que se conoce como **inversión socialmente responsable**. Este comportamiento a su vez fuerza a las empresas a un mayor compromiso con los temas medioambientales, éticos y sociales. Es por ello por lo que este tipo de inversiones empiezan a suponer una parte importante dentro de las carteras de los inversores y con total seguridad esta tendencia seguirá creciendo en los próximos años.

En los siguientes gráficos de Morningstar y BlackRock se muestran las expectativas de crecimiento que se esperan para los próximos años en este tipo de productos.

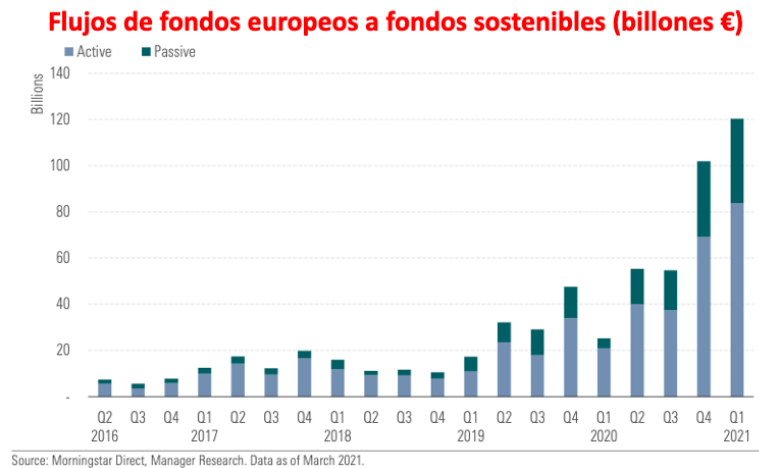


Figure 3-3-1: Flujos de fondos sostenibles. Morningstar (March 2021)

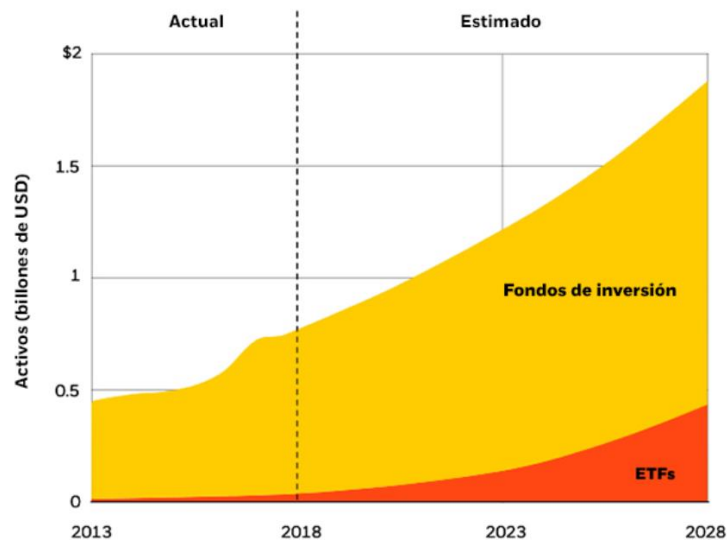


Figure 3-3-2: BlackRock estimacion de activos de Fondos y ETFs sostenibles

3.2.1 RSI : INVERSIÓN SOCIALMENTE RESPONSABLE

Este nuevo concepto de inversión surgió en el **siglo XX en Estados Unidos**, asociada a entidades religiosas que en sus inversiones excluían a empresas que no consideraban éticas como pueden ser las relacionadas con el tabaco, el alcohol o las armas. Pero es a finales de los 60 cuando está se desarrolla totalmente y encuentra su mayor impulso tras la crisis de 2008.

Esta modalidad de inversión aplica los criterios conocidos como **ESG** (*Enviromental, Social, Governance*) a la hora de seleccionar las empresas.

Existen diferentes estilos de inversión dentro de este campo:

- **Exclusión**: consiste en **evitar la inversión** en compañías o sectores que van **directamente en contra** de los criterios ESG, como pueden ser la industria armamentística, tabaquera, que emplean experimentación animal...
- **Mejor de su clase**: En este caso no se descartan sectores, pero se escogen aquellas compañías que presentan un **mayor compromiso con las temáticas ESG** dentro de cada sector.
- **Integración ESG**: **Integra los criterios ESG en el proceso de análisis** y selección de valores.
- **Inversión con temática en sostenibilidad**: comprende estrategias **centradas en un tema específico** que tiene en cuenta de manera más o menos estricta los criterios ESG. (Ejm: gestión de agua, envejecimiento de la población ...)
- **Inversión de impacto**: consiste en medir como la cartera de **inversión aporta valor** en cuestiones como, por ejemplo, medioambientales, o que por ejemplo llevan a cabo algún tipo de proyecto social. En este caso el principal objetivo no es el retorno financiero.

	Descripción	Valor financiero
Exclusión	Exclusión de industrias o sectores con temáticas contrarias a los criterios ESG.	La rentabilidad es el objetivo principal
Screening Positivo/mejor de su clase	Inversión en empresas con las mejores puntuaciones ESG en relación con sus homólogos	Puede ayudar a evitar riesgos altos en comparación con sus homólogos.
Integración ESG	Inclusión explícita de criterios ESG en el proceso de de análisis y selección de compañías.	La rentabilidad es el objetivo principal
Inversión temática	Selección de activos con temáticas de sostenibilidad o sociales	La rentabilidad es uno de los objetivos principales junto con la integración ESG
Inversión de Impacto	inversión con objetivos dirigidos a resolver problemas sociales o medioambientales	El rendimiento tiene un papel secundario aunque si importante.

Tabla 3-4:Tipos de inversión socialmente responsable

3.3 MACHINE LEARNING, DEEP LEARNING, ARTIFICIAL INTELLIGENCE AND DATA SCIENCE.

Todos estos nombres se escuchan actualmente por todos lados, ¿Pero ¿qué son exactamente? ¿Cuál es realmente la diferencia entre ellos? ¿Cómo funcionan? Y ¿Cuál es su utilidad?

Esta son las preguntas que trataremos de responder en este apartado de forma que estos conceptos queden claros y permitan seguir el desarrollo del proyecto sin dificultades.

Los conceptos de “machine learning”, ”Deep learning”, ”Artificial Inteligencia”, ”Data Science” pueden resultar bastante confusos. Por ello con el fin de aclarar esta confusión, en la figura de abajo se puede observar claramente las relaciones que existen entre ellos.

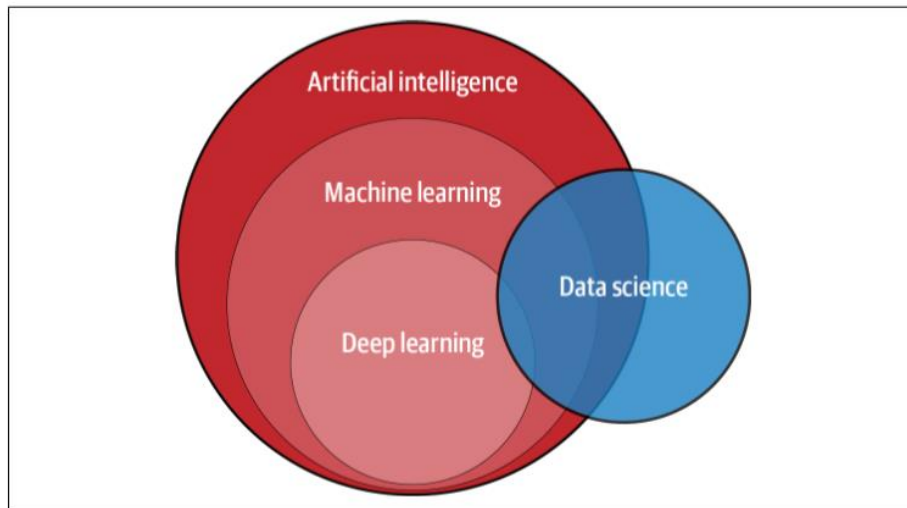


Figure 3-3: AI, Machine learning, deep learning and data science. (libro “Machine Learning and Data Science for Finance”.)

Como se puede ver el término “Machine learning” es una subcategoría de la Inteligencia Artificial y a su vez “Deep learning” es un subconjunto dentro de “Machine learning”.

Por otro lado, *Data Science* no es exactamente un subconjunto de la Inteligencia Artificial, pero esta hace uso de *machine learning*, *Deep learning* e *Inteligencia Artificial* para analizar y establecer conclusiones a partir de grandes cantidades de datos.

Al igual que vimos con la inversión sostenible, el crecimiento de estas tecnologías dentro de la industria financiera está siendo muy significativo en los últimos años.

En la siguiente imagen pueden verse unas predicciones que se hicieron en el año 2018 del crecimiento que tendría la Inteligencia Artificial en el sector Bancario y financiero.



U.S. Artificial Intelligence in BFSI Market Share, By Solution, 2017 (USD Million)

Figure 3-4: <https://www.globalbankingandfinance.com/artificial-intelligence-ai-in-bfsi-market-to-grow-at-30-cagr-from-2018-2024/>

3.3.1 MACHINE LEARNING

En este apartado se ofrecerá una pequeña introducción acerca de las diferentes técnicas de *Machine learning* que existen, ya que son estas de las que se hará uso para el desarrollo de la herramienta.

A continuación, se puede ver en la imagen los diferentes tipos y subgrupos que existen dentro de esta disciplina:

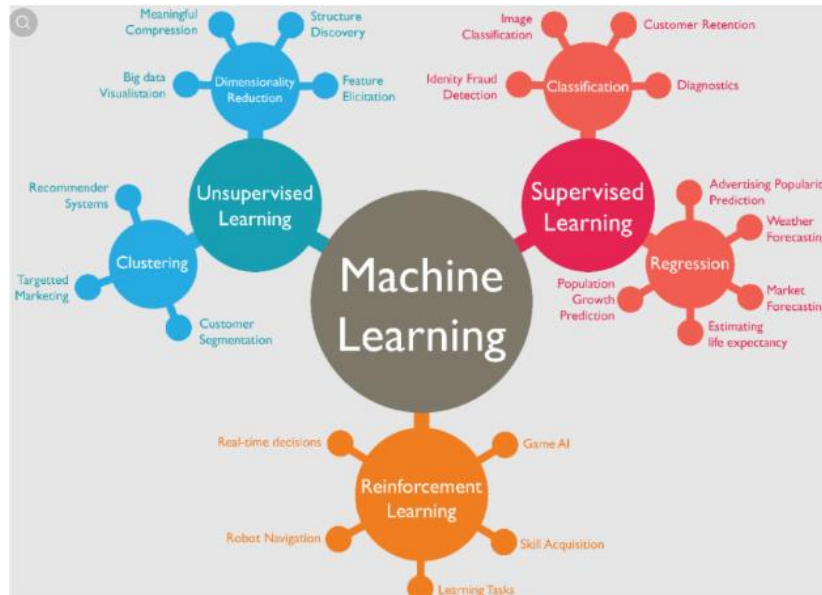


Figure 3-5:Tipos dentro de Machine learning(<https://www.freegyann.com/post/build-a-model-with-python>)

En la siguiente página, se presenta un esquema que se ha creado con el fin de ilustrar las principales técnicas que existen dentro de la minería de datos. Algunos de los cuales emplearemos en el desarrollo de nuestro prototipo:

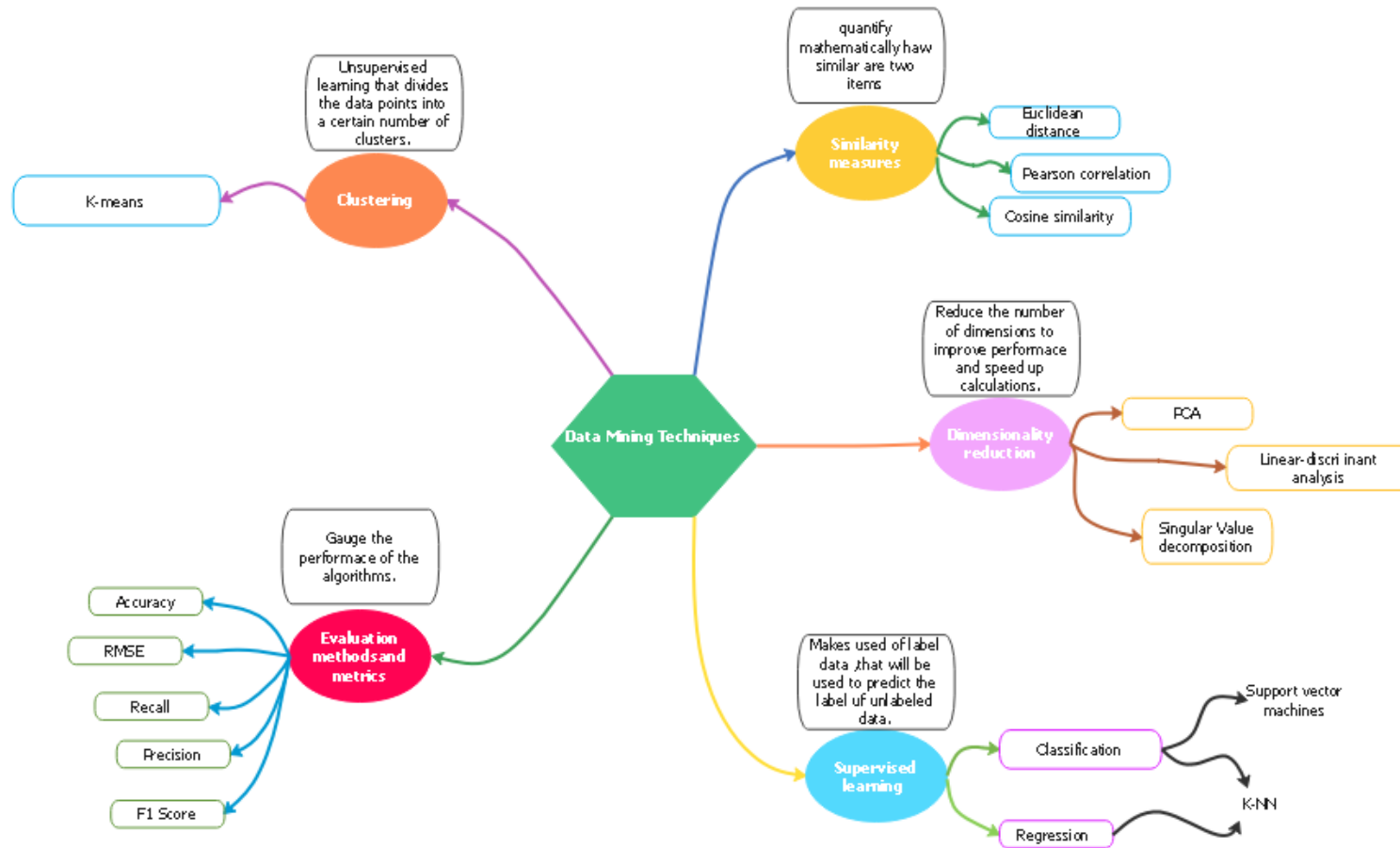


Figure 3-6 :Mapa mental técnicas de Machine learning.

Aprendizaje supervisado: El objetivo es entrenar un modelo mediante de datos etiquetados que permita hacer predicciones de datos futuros. Se parte de muestras donde la **salida deseada es conocida** (etiquetas), es decir existe un conocimiento a priori. Dentro de este tenemos dos tipos de algoritmos: **Regresión** y **Clasificación**.

- **Regresión:** Se emplea para predecir el valor de la variable de salida, cuando esta es **continua**. Se parte de una serie de variables explicativas y se predice el valor de la variable de salida de tipo continuo. Se trata por lo tanto de establecer la relación que existe entre las variables de entrada y salida.
- **Clasificación:** En este caso el objetivo es predecir la clase **categorica** de nuevas entradas basándonos en observaciones pasadas. En este caso la salida es por lo tanto una variable **discreta** y no continua como en el caso de Regresión.

Aprendizaje no supervisado: Aprendizaje automático usado para sacar conclusiones a partir de datos donde **no existe un conocimiento a priori**, lo que significa que los datos de entrada no están etiquetados.

Dentro de este encontramos dos tipos: **Reducción de la dimensionalidad** (*Dimensionality Reduction*) y **Agrupación** (*Clustering*).

- **Reducción de la dimensionalidad:**
Consiste en un proceso que **reduce el número de variables**, pero asegurando que la **información relevante se conserva**, así como el desempeño del modelo.
- **Clustering:**
Permite **agrupar los datos por similitud** en grupos o conjuntos, donde los miembros del mismo grupo tengan características similares. Se trata de descubrir estructuras escondidas en los datos.

Aprendizaje reforzado: El objetivo es que el algoritmo aprenda a tomar decisiones, mediante **recompensas o castigos**. De forma que se tomen aquellas decisiones que permitan **maximizar la recompensa** en una situación dada. El sistema toma una decisión en una situación dada y aprende si dicha acción fue correcta o no según la recompensa recibida y va aprendiendo a través de la experiencia.

Capítulo 4. DESARROLLO TEÓRICO

4.1 INTRODUCCIÓN

El objetivo de este proyecto es la creación de un modelo de aplicación web de un Robo-Advisor pero que tiene algunas diferencias y ventajas sobre los RA tradicionales.

En este capítulo se hará una revisión de los diferentes procesos involucrados en estos modelos actuales de asesoramiento automatizado como son el **perfilado del cliente** y la **colocación de activos**. También se hará referencia a algunos de los diferentes modelos y alternativas que existen actualmente para llevar a cabo estos procesos.

Finalmente se hace un análisis de los **sistemas de recomendación** y como podría ser su implementación en los Robo-Advisors, en este caso enfocado a temática ESG.

Cabe destacar que este prototipo que se pretende implementar tiene algunas diferencias con los modelos tradicionales como son:

1. Los asesores automatizados cuentan con **carteras modelo** predefinidas que se asignan a los clientes en función de los resultados ofrecidos por el test de idoneidad del que se hablará más adelante. Para nuestro modelo queremos que el cliente sea quien elija los activos que formarán su cartera y por lo tanto la colocación de activos y **las carteras de cada cliente serán diferentes**.
2. Debido a la no utilización de carteras modelo la tarea de colocación de activos ha presentado algunas complicaciones. Es por ello por lo que para no elevar en exceso la complejidad de la aplicación se ha optado por ofrecer a cada cliente el porfolio más eficiente, pero sin atender a su perfil de riesgo. Es decir, aunque la aplicación si contará con **un test de idoneidad, éste será de carácter ilustrativo** y no se empleará para el perfilado del cliente.

3. Otra de las características que diferencia nuestro modelo de la mayoría de los que existen actualmente es el sistema de **recomendación de ETFs ESG**, de forma que se ofrecerá a los usuarios recomendaciones que se adapten a sus gustos e intereses en cuanto a la inversión responsable se refiere.

4.2 PERFILADO DEL CLIENTE

En el capítulo 3 se habló del funcionamiento de los Robo-Advisors y de cómo estos a través de un cuestionario de tan solo 10 preguntas asigna un perfil de riesgo al cliente. Pero ¿Requiere este proceso de Inteligencia Artificial?

Realmente, se trata de un problema bastante sencillo donde a través de las respuestas del cliente podemos asignarle un perfil sin necesitar de ningún tipo de algoritmo inteligente o modelo de machine learning. Es por ello por lo que la mayoría de Robo-Advisors no son inteligentes, es decir no hacen uso de la Inteligencia Artificial.

Es decir, para la asignación de la tolerancia al riesgo del cliente la mayoría de los procesos se basan en un cuestionario a partir del cual **se asigna un valor riesgo beneficio y en función de este una distribución de cartera**. Por lo que no es necesaria ninguna inteligencia por parte del algoritmo.

4.2.1 TEST DE IDONEIDAD (MIFID)

MIFID (Market in Financial instruments Directive) Es la directive de la Unión Europea sobre instrumentos financieros que busca defender y proteger al inversor, además de fomentar la competencia y transparencia de información de productos de inversión para adecuarse al perfil de riesgo del inversor.

El **test de idoneidad** MIFID y tiene como objetivo obtener información acerca de los **objetivos** de inversión, la **situación** financiera y los **conocimientos** sobre los mercados financieros del cliente, así como su **aversión al riesgo**. Esta prueba se exige a todos los

clientes minoristas a los que se les vaya a prestar un servicio financiero y tiene como objetivo garantizar la protección del inversor.

Con esta información se determina el perfil de riesgo de este. Se suele establecer la siguiente clasificación.

1. **Conservador**
2. **Moderado**
3. **Arriesgado.**

No todos los test de idoneidad son iguales, cada entidad tiene el suyo. A continuación, se encuentran los enlaces a varios de estos test.

Expansión: <https://www.expansion.com/ahorro/2015/10/06/5613bcd0ca4741cc7d8b45e1.html>

Indexa Capital: <https://indexacapital.com/es/esp/questions>

Vanguard: <https://investor.vanguard.com/calculator-tools/investor-questionnaire/>

Para nuestra la creación del prototipo se han escogido las siguientes preguntas:

- **Para evaluar el punto de partida del cliente:**
 1. En cuanto a la inversión en Fondos, ETFs o acciones individuales me describiría a mí mismo como...
 - a) Muy inexperto
 - b) Algo inexperto
 - c) Algo experimentado
 - d) Experimentado
 - e) Muy experimentado
 2. Mis actuales y futuras fuentes de ingresos son ...
 - a) Muy inestables
 - b) Inestables.
 - c) Algo estables
 - d) Estables

e) Muy estables.

**Cuanto más estables sean las fuentes de ingresos y la estabilidad financiera del usuario mayor riesgo puede asumir su cartera.*

3. ¿Qué edad tienes?

4. ¿Qué porcentaje de tus ingresos se van en gastos?

- a) Menos del 25%
- b) 25-50%
- c) 50-75%
- d) Más del 75%

5. ¿Cuál es su nivel de estudios?

- a) Enseñanza obligatoria
- b) Formación profesional/ grado
- c) Educación universitaria o superior

• **Para evaluar las metas y objetivos del cliente:**

1. ¿Qué rentabilidad buscas para tu cartera de inversión?

- a) 1-2% anual. Protegerme frente a la inflación y estabilidad.
- b) 3-4% anual. Equilibrio entre siego y rentabilidad.
- c) 5-6% anual. Con un mayor nivel de riesgo.
- d) La máxima rentabilidad posible sin importar el riesgo.
- e) No me importa la rentabilidad, pero quiero no quiero perder mi dinero.

• **Para determinar el periodo de inversión:**

1. Planeo empezar a recoger el dinero de mis inversiones en ...

- a) 1 año o menos
- b) 1-2 años
- c) 3-5 años
- d) 6-10 años
- e) 11-15 años
- f) Más de 15 años.

- **Para determinar el riesgo que el cliente está dispuesto a asumir:**

1. Imagina que su cartera de inversiones pierde un 10% de su valor en un mes ¿Qué harías?
 - a) Vender
 - b) Vender una porción
 - c) Mantener
 - d) Comprar más.

**El objetivo es identificar la aversión al riesgo del usuario y como afrontaría fuertes pérdidas en su portfolio. Cuanto mayor sea su tolerancia al riesgo más arriesgada puede ser su cartera.*

2. Durante las caídas del mercado tiendo a vender partes de mis activos más arriesgados e invertir en activos más seguros.
 - a) Totalmente en desacuerdo
 - b) En desacuerdo
 - c) Algo de acuerdo
 - d) De acuerdo
 - e) Totalmente de acuerdo
3. Si piensas en la palabra “Riesgo” ¿Cuál de las siguientes palabras te viene a la mente?
 - a) Pérdida
 - b) Incertidumbre
 - c) Oportunidad
 - d) Emoción

4.2.1.1 ¿Cómo se determinar el perfil de riesgo del inversor en base a sus respuestas?

En base a los resultados de la prueba se asigna un valor del 1 al 10 al perfil de riesgo de la inversión y en base a este se establece una proporción de **renta variable/ renta fija**.

Por ejemplo, se establecen 5 composiciones de carteras diferentes:

- Riesgo: Bajo
- Riesgo: Medio-Bajo
- Riesgo: Medio
- Riesgo: Medio-Alto
- Riesgo: Alto

El riesgo de la cartera determinará tanto la ratio de inversión renta fija/ Renta variable como el tipo de activos en los que se invierte. Siendo los más volátiles los más arriesgados.

Dado que se ha hecho referencia a la regulación europea MIFID y dado que este proyecto tiene un enfoque sostenible se quiere hacer mención a la nueva regulación MIFID II ,bajo esta normativa la entidades financieras serán responsables no solo de proporcionar el rendimiento financiero correspondiente al perfil de riesgo del inversor, sino también del nivel esperado de impacto social que coincida con las preferencias de sostenibilidad de los clientes.

Lo que ilustra claramente la tendencia del sector y su interés por incorporar en sus productos criterios ESG y la importancia de proporcionar a los clientes esta información de forma transparente y clara.

4.2.1.2 ¿Podemos emplear IA y ML para establecer el perfil de riesgo del cliente?

En esta sección se expondrán y analizarán **dos ejemplos** de métodos que emplean técnicas de machine learning para identificar o clasificar a los clientes según su perfil de riesgo, a fin de ilustrar algunas posibilidades para la integración de dichas tecnologías en este proceso.

Ambos ejemplos han sido obtenidos del libro “*Machine Learning and Data Science for Finance*”.

Ejemplo 1: Regresión supervisada para la valoración del perfil de riesgo.

Está demostrado que los inversores están sesgada a la hora de determinar su tolerancia al riesgo, especialmente en momento difíciles del mercado. En el libro “*Machine Learning and Data Science for Finance*” proponen un ejemplo de cómo podría usarse la inteligencia artificial para **determinar la tolerancia al riesgo real de los clientes** a partir de una base de datos.

El algoritmo de machine learning que ellos proponen se basa en un modelo de aprendizaje supervisado de regresión. Consiste en lo siguiente:

1. Parten de **la base de datos de la Reserva Federal *Survey of Consumer Finances*(SCF)**.

En esta base de datos se recoge información acerca de la riqueza, activos financieros y no financieros y datos demográficos de los hogares, de un grupo de individuos entre los periodos de 2007 (previo a la crisis) y 2009 (después de la crisis). El objetivo es ver como la colocación de activos de los inversores cambió durante la crisis de 2008 y con ello evaluar **cómo se comportaron realmente cuando las condiciones del mercado eran complicadas**.

2. En primer lugar, se calcula el valor de los activos **con riesgo** (fondos, acciones y bonos) y los activos **sin riesgo** (cuentas de ahorro, depósitos y otros equivalentes) de cada individuo.
3. Como medida de la tolerancia al riesgo individual se calcula el **cociente** entre los activos con riesgo y el total de activos.
4. Compara la variación de la tolerancia al riesgo de los individuos en los dos periodos, previo a la crisis y posterior a la crisis. Establece como medida fiable a aquellos que solo hayan experimentado una variación inferior al 10 % en su tolerancia al riesgo.

Crea una nueva variable llamada “True risk tolerance” que es la media de la tolerancia al riesgo de estos inversores en los dos periodos. Esta será la variable a predecir en el estudio.

5. Se asigna a la variable “Tolerancia real al riesgo ” el valor de la media de la tolerancia de estos inversores en los dos periodos.
6. De las 500 variables de la base de datos se escogen aquellas que se considera que tiene mayor influencia en la tolerancia al riesgo y serán las que se usen como variables predictivas.

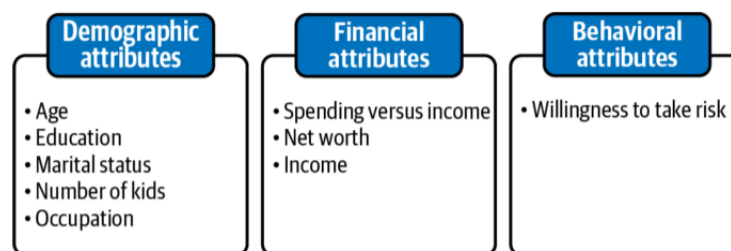


Figure 4-1: Variables predictivas (Libro “Machine Learning and Data Science Blueprints for Finance”)

7. Dividimos la base de datos en “Train” y “Test”. Asignando el 80% de los datos al primer conjunto. Para medir el funcionamiento del algoritmo se usa R^2 y se emplean 10 como número de validaciones cruzadas.
8. Tras evaluar diferentes modelos el que mejor resultado ofrece, obtiene los siguientes valores:

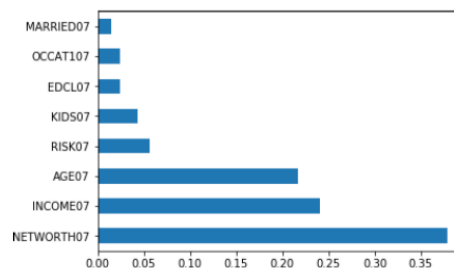


Figure 4-2: Influencia de las variables en el modelo (Libro “Machine Learning and Data Science Blueprints for Finance”)

Donde el eje X representa la magnitud de influencia de cada variable en la obtención del valor de la variable de salida (tolerancia real al riesgo).

9. El resultado del modelo arroja un valor entre 0 y 100 en base al cual se calcula la colocación de activos.

Para el cálculo de esta colocación emplea el método tradicional “mean-variance optimization ” que se explicará en detalle en la sección de colocación de activos.

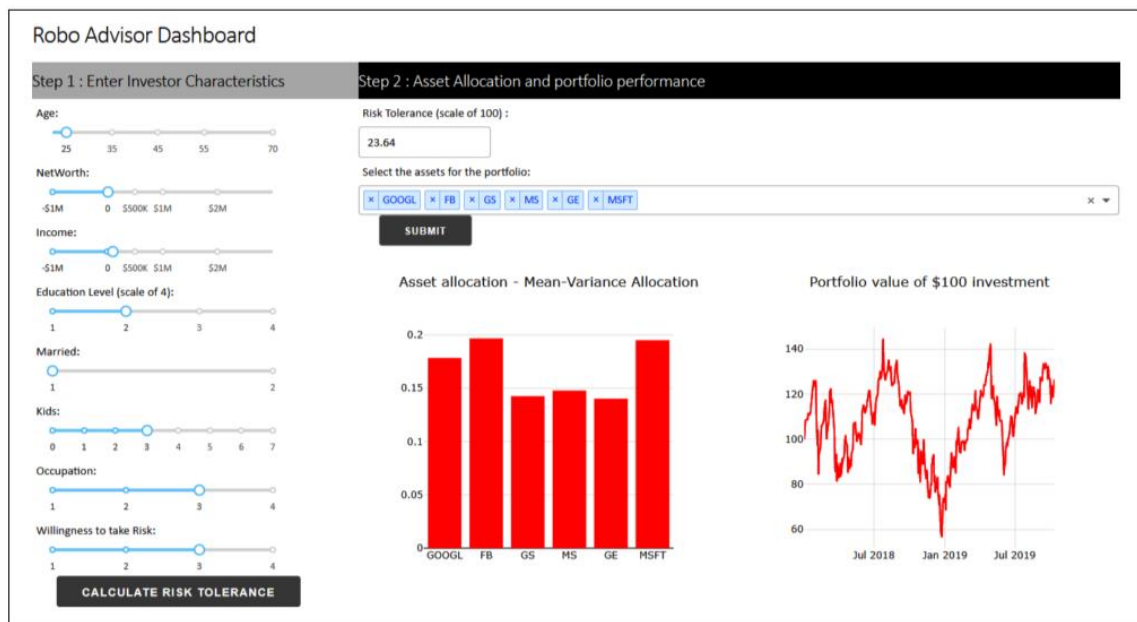


Figure 4-3: Resultado final del modelo (Libro “Machine Learning and Data Science Blueprints for Finance”)

Ejemplo 2: Clustering Investors.

El objetivo de este algoritmo es identificar aquellos **inversores que cumplen características similares** y similar disposición para asunción de riesgos.

El algoritmo en base a una serie de variables establecerá una serie de grupos, donde cada grupo representa a un conjunto de inversores que tienen características similares.

Esta agrupación se hace a partir de factores como la edad, ingresos, tolerancia al riesgo...

La base de datos que se usa en este caso también es obtenida de “Survey of Consumer Finances” de la Reserva Federal. La misma que se usó en el ejemplo anterior.

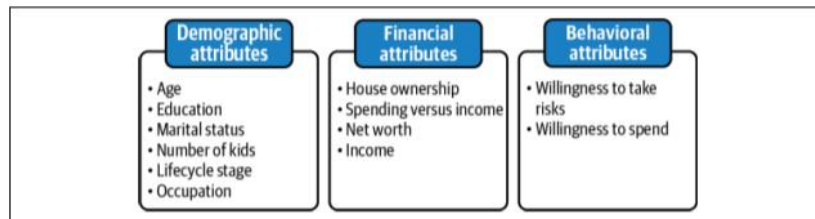


Figure 4-4: Atributos para la agrupación de individuos. (libro “Machine Learning and Data Science Blueprints for Finance”)

El algoritmo de machine learning que se va a emplear es “K-means Clustering” y para la selección del número de clústeres óptimo se emplean:

- SSE (Sum of squared errors)
- Silhouette score.

Una vez se ha determinado el número de grupos se evalúan las características de cada uno de ellos para extraer conclusiones.

A partir de los valores medios de los atributos para cada grupo podemos inferir la capacidad de riesgo de cada uno de los grupos.

Los resultados obtenidos son los siguientes:

Cluster	Features	Risk capacity
Cluster 0	Low age, high net worth and income, less risky life category, willingness to spend more	High
Cluster 1	High age, low net worth and income, highly risky life category, willingness to take risk, low education	Low
Cluster 2	High age, high net worth and income, highly risky life category, willingness to take risk, owns home	Medium
Cluster 3	Low age, very low income and net worth, high willingness to take risk, many kids	Low
Cluster 4	Medium age, very high income and net worth, high willingness to take risk, many kids, owns home	High
Cluster 5	Low age, very low income and net worth, high willingness to take risk, no kids	Medium
Cluster 6	Low age, medium income and net worth, high willingness to take risk, many kids, owns home	Low

Figure 4-5: Conclusiones algoritmo. (Libro: “Machine Learning and Data Science Blueprints for Finance”)

4.3 COLOCACIÓN DE ACTIVOS

Existen varios modelos para la colocación de activos que permiten establecer los pesos de los activos de una cartera para que se ajusten al perfil de riesgo del inversor.

El objetivo de estos es colocar el capital en diferentes clases de activos de forma que se maximice los retornos en función del riesgo asumido. Es la **volatilidad** la que se toma como **medida de riesgo** de un activo dado.

En este apartado hablaremos de tres de ellos, el primero de ellos el tradicional método “**mean-variance optimization**” (MVP) y los otros dos basados en algoritmos que emplean machine learning, también sacados del libro “*Machine Learning and Data Science for Finance*”. El primero de ellos se basa en un modelo de **aprendizaje no supervisado** basado en una técnica de reducción, en concreto PCA (Principal component analysis) y el segundo se basa en la técnica de **clustering** para la clasificación de los diferentes perfiles de riesgo.

4.3.1 HISTORIA DE LA OPTIMIZACIÓN DE CARTERAS

La optimización de los portfolios comenzó con **el Doctor Harry Markwitz** quien creó lo que conocemos como “**Mordetn Portfolio Theory**”, de aquí se obtiene que la curva llamada **Effiecient Frontier** es donde se encuentran las carteras eficientes, aquellas que tienen el mayor retorno para un grado de riesgo dado.

Sin embargo, esta teoría tiene sus fallos, el primero de ellos es que requiere de la **estimación de los retornos esperados** y obviamente no se sabe con certeza cuales serán estos retornos por lo que al hacer esta aproximación introducimos un error importante. Por otro lado, el proceso matemático requiere del cálculo de la **inversa de la matriz** de covarianzas lo que se complica si esta matriz es muy grande y se vuelve bastante inestable de forma que pequeños cambios en la varianza pueden derivar en cambios significativos.

A raíz de esto surgió el modelo de **Black Litterman** para resolver algunos de los problemas del modelo anterior. En este caso no necesitas predecir los retornos futuros lo que permite calcular portafolios mucho más precisos.

El siguiente avance que surgió fue **Risk Parity** que cambia el enfoque del problema. Hasta ahora los portafolios se diversificaban con el capital, con esta nueva visión lo que se **diversifica es el riesgo**.

Finalmente llegamos a lo que se conoce como **HRP** desarrollado por el doctor **Marcos López de Prado** en 2019. El concepto detrás de este método es que los activos se agrupan según su parecido y después se calcula la colocación del capital entre estos grupos. Este método tiene una ventaja frente a los otros ya que **no requiere de invertir la matriz** de covarianza. Esta técnica genera portafolios mucho más diversificados para una para un riesgo muy similar.

4.3.2 MARKOWITZ MODEL: MEAN-VARIANCE OPTIMIZATION

También conocido como “Modren Portfolio Theory”, es una teoría de inversión que estudia como maximizar el retorno de una cartera a la vez que se minimiza el riesgo de la misma, mediante la selección de los pesos de los activos que compondrán la cartera.

Como medida del riesgo se emplea la volatilidad y la cartera se conforma según la tolerancia al riesgo de cada individuo, seleccionando aquella que resuelta en el mayor retorno para dicho nivel de volatilidad.

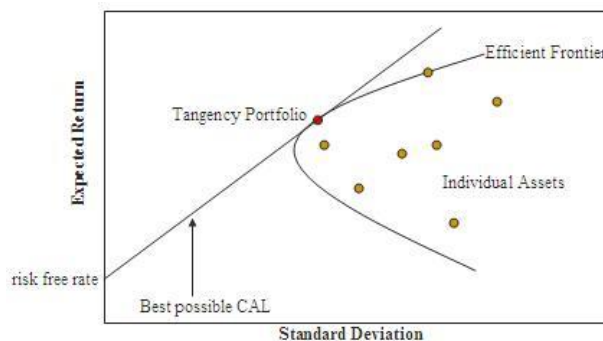


Figure 4-6: Efficient Frontier (Wikipedia)

En la imagen superior “Tangency Portfolio” hace referencia a la cartera eficiente con mayor valor de “Sharpe Ratio”, donde el **Sharpe Ratio** es el cociente entre el retorno esperado y la volatilidad.

Por lo tanto, la construcción de las carteras se ajusta a las expectativas y perfil de riesgo del inversor. Tratando de escoger activos que estén poco correlacionados entre si obteniendo de esta forma una mayor diversificación para reducir la volatilidad de la cartera.

De esta forma a cada inversor se le ofrece la cartera óptima, que es aquella que determina la tangencia entre **la frontera eficiente** y la **curva de indiferencia del inversor** en concreto, como se ve en la imagen inferior.

Las curvas de indiferencia representan las diferentes combinaciones de rentabilidad-riesgo que le otorgan la misma utilidad al inversor y por lo tanto le son indiferentes.

Cada punto de la curva de indiferencia representa una combinación diferente de retorno y riesgo que satisface el nivel de riesgo con el que el inversor se siente cómodo. El punto óptimo se encuentra en el punto de tangencia entre la frontera eficiente y la curva de indiferencia.

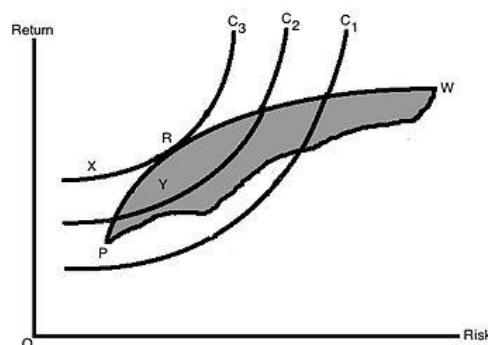


Figure 4-7: Efficient Frontier y curvas de indiferencia (Wikipedia)

Dado que esto no se incluirá en la creación del prototipo no se indagará en los aspectos matemáticos de este problema de optimización.

4.3.3 EIGEN -PORTFOLIO

Este es un ejemplo de una metodología para la colocación de activos basada en una técnica de machine learning.

MPT emplea los retornos esperados de estos activos para hacer sus estimaciones. La naturaleza errática y dificultad de predicción de estos retornos pueden provocar errores en las estimaciones.

Empleando PCA podemos coger **la matriz de covarianzas $n \times n$** y crear un **n portfolios principales linealmente independientes**. Podemos usar los componentes principales estandarizados como pesos de la cartea con la garantía estadística de que sus retornos no están linealmente correlacionados.

A continuación, se presenta el ejemplo extraído del libro “*Machine Learning and Data Science for Finance*” con el objetivo de ofrecer una mejor comprensión de todo el proceso y como la descomposición en *egien*vectors de la matriz de covarianzas de los retornos puede ayudar en las decisiones de inversión.

El funcionamiento de este modelo es el siguiente:

1. La base de datos que se emplea en el modelo son los precios de cierre de los activos que formarán parte de la cartera. En el ejemplo del libro emplean el Dow Jones Industrial Average, que contiene 30 compañías.
2. Observamos la correlación de los retornos de los diferentes activos con **una matriz de correlación**.
3. Entrenamos el modelo **PCA** .
4. Los **resultados** que se obtiene en el libro del cual se ha extraído este ejemplo son:

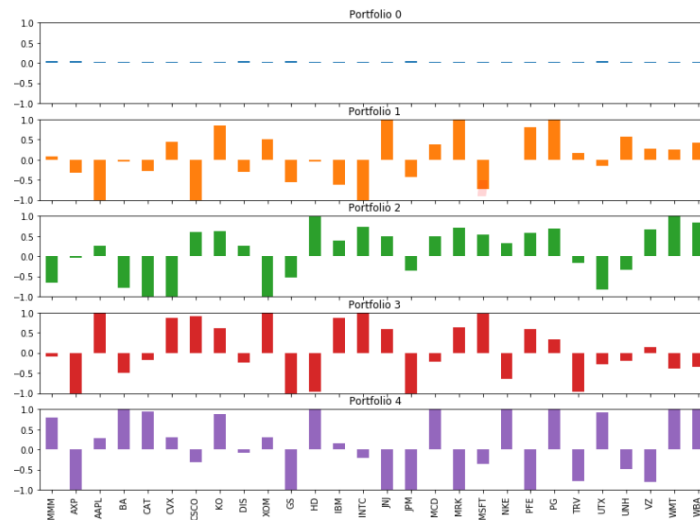


Figure 4-8: Diferentes Eigen Portfolios (libro “Machine Learning and Data Science Blueprints for Finance”)

Los *egien*vectores se corresponden con cada uno de los *egien*portfolios. Cada barra representa el peso asociado a cada acción y en el caso de que las barras sean negativas representan una posición en corto en esa acción.

La idea detrás de esto es que cada *egien*-portfolio **representa algún tipo de factor de riesgo independiente**. El portfolio principal es a que, con mayor varianza, típicamente el **riesgo sistemático del mercado** y el resto de los portfolios suelen corresponder a factores del sector o la industria.

El siguiente paso en el ejemplo del libro es encontrar el mejor de los *egien*portfolios. Para ello lo que hace es calcular y comparar los valores de **sharpe-ratio**. Y los resultados que obtiene son los siguientes:

	Return	Vol	Sharpe
0	0.115	0.133	0.862
7	0.096	0.693	0.138
5	0.100	0.845	0.118
1	0.057	0.670	0.084

Figure 4-9: Resultados de los diferentes porfolios.

El porfolio con mejor ratio es el *Portfolio 0*, con la siguiente distribución de activos:

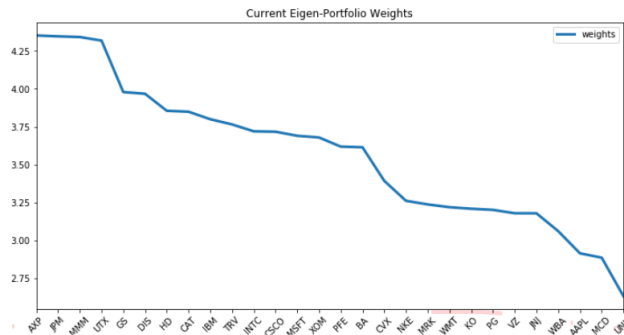


Figure 4-10: Distribución de activos Portfolio 0 (libro “Machine Learning and Data Science Blueprints for Finance”)

4.3.4 HIERARCHICAL RISK PARITY

En “Building Diversified Portfolio That Outperform Out-of-Sample” de Marcos López de Prado, el autor propone un método para la colocación de activos basada en Clustering llamada “**Hierarchical risk parity**”.

Como vimos en la introducción **Risk parity**, Es una estrategia de colocación de activos que usa el riesgo para determinar la colocación entre los diferentes componentes de una cartera. Esta estrategia modifica la tradicional teoría del portfolio moderna (MPT) .

Este método puede dividirse en 3 pasos diferenciados:

1. **Tree clustering:** Consiste en agrupar los diferentes activos basándose en su correlación. Cada *cluster* será dividido a su vez en *clusters* más pequeños hasta llegar a cada uno de los activos. Definiendo así una jerarquía que ayuda a resolver los problemas de estabilidad que surgen al intentar invertir la matriz de covarianzas.

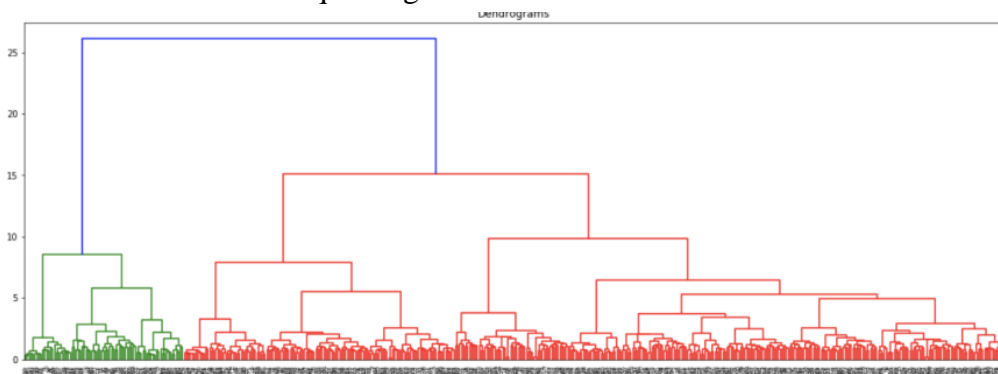


Figure 4-11: Dendrogram (Libro “Machine Learning and Data Science Blueprints for Finance”)

2. Quasi-diagonalization: Consiste en reorganizar la matriz de covarianzas de forma que los activos similares queden juntos. En la imagen inferior se muestra un ejemplo de cómo sería la matriz de covarianzas antes y después de la aplicación de este paso.

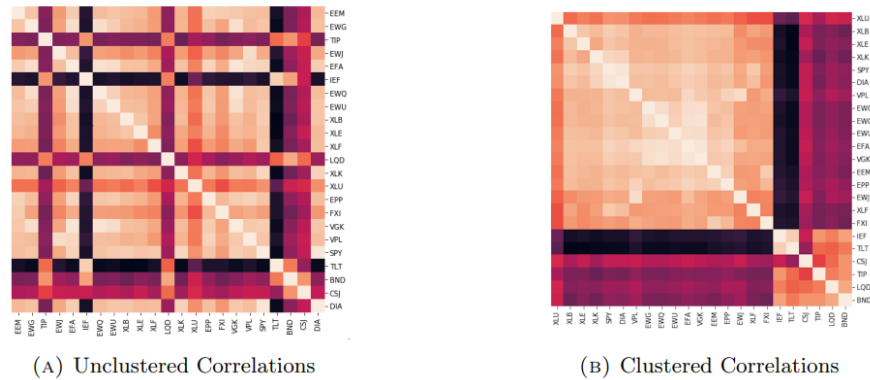


Figure 4-12: Quasi-diagonalization (libro “Machine Learning and Data Science Blueprints for Finance”)

3. Recursive bisection: Este último paso se deciden los pesos del portfolio mediante un método recursivo basado en las covarianzas de los clusters .Se calcula la volatilidad individual de cada uno de los *clusters* y se actualizan los pesos de forma inversamente proporcional a sus volatilidades.

En la imagen inferior se muestra un ejemplo de la colocación de activos empleando los dos métodos y vemos como HRP da como resultado un portafolio mucho más diversificado.

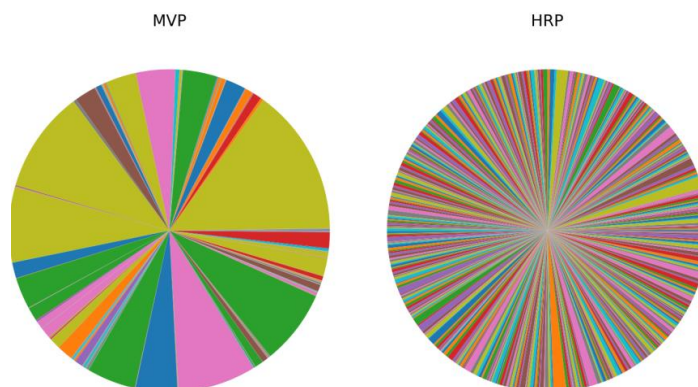


Figure 4-13: Colocación activos MVP vs HRP(libro “Machine Learning and Data Science Blueprints for Finance”).

4.4 SISTEMAS DE RECOMENDACIÓN

El sistema de recomendación de nuestro prototipo pretende ser capaz de ofrecer recomendaciones de inversión sostenible de forma personalizada.

Para ello en este apartado se empezará por presentar los principales modelos de recomendación que se usan habitualmente, analizando sus principales ventajas y desventajas para finalmente elegir aquellos que más se ajustan a nuestro objetivo.

También se mencionarán las principales complicaciones que se han encontrado a la hora de decidir y diseñar el algoritmo de recomendación.

4.4.1 EL PROBLEMA DE RECOMENDACIÓN

Podemos dividir el problema de las recomendaciones en dos tipos diferentes: **Ranking** y **Predicción**.

En el siguiente esquema se muestra en que consiste cada uno:

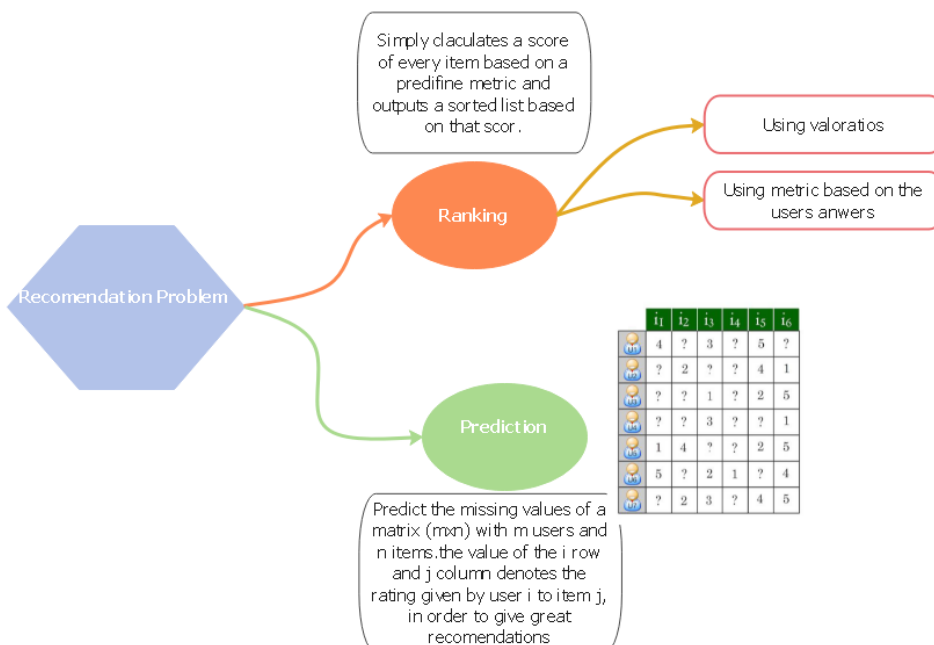


Figure 4-14: Esquema problema de recomendación.

4.4.2 TIPOS DE SISTEMAS DE RECOMENDACIÓN

Un sistema de recomendación es aquel que predice las valoraciones o preferencias que un usuario pueda tener a cerca de un determinado producto. Es decir, recomienda productos o servicios al usuario basándose en su comportamiento anterior y el comportamiento de otras personas.

El sistema tratará de comprender por lo tanto cuales son las preferencias y gustos del usuario. Para ello puede valerse de feedback específico (valoraciones que ha hecho el usuario explícitamente) o comportamiento implícito (Ej. Clicks que hace el usuario, si ha comprado o no un determinado producto, cantidad de tiempo que ha pasado viendo un video ...)

El tema de los sistemas de recomendación es bastante amplio, por lo que para este proyecto nos centraremos en presentar las bases fundamentales y principales sistemas que existen yendo de más simple a más complejo.

➤ Sistemas no personalizados:

Este tipo de recomendaciones no tienen en cuenta las características del usuario. Hacen una **recomendación general para todos los usuarios**, como pueden ser recomendaciones de expertos o el Top 10 de canciones escuchadas en un país.

▪ TOP N:

Dentro de este sistema presentaremos dos opciones:

1. **No tiene en consideración la popularidad** de la película. Es decir, si un ETF ha sido valorado con una puntuación de 9 por 10000 usuarios tiene el mismo peso que otro que ha sido valorado con una puntuación de 9 por solo 10 usuarios.
2. En este caso se introduce y **ponderan las puntuaciones en base a su popularidad.**

A continuación, mostramos un ejemplo de cómo sería la implementación de un algoritmo de este estilo para un caso como el nuestro:

Los pasos que seguir serían:

1. Elegir la métrica en función de la cual se evalúan los ETFs.
2. Calcular la nota de cada ETF .
3. Mostrar la lista de los ETFs en orden decreciente de notas.

La métrica que se usaría para crear el ranking serán las valoraciones de los clientes para los distintos ETFs.

Además, como es lógico, no tiene la misma importancia una calificación respaldada por 10 personas que por 1000 , por lo que debemos **ponderar estas calificaciones según su popularidad.**

Para ello se emplea la siguiente fórmula:

$$Nota\ ponderada = \left(\frac{v}{v + m} * R \right) + \left(\frac{m}{v + m} * C \right)$$

Ecuación 4-1: Ecuación Nota ponderada

v: número de votos del ETF.

m: número mínimo de votos requeridos para que el ETF aparezca en el ranking.

Cuanto mayor sea el valor de *m* mayor será el énfasis que se hace en la importancia de la popularidad. Es decir, si establecemos un valor de 0.8 implica que para que un ETF aparezca en la lista del Top 10 debe tener más votos que el 80% de los ETFs de la base de datos.

R: media de las votaciones del ETF.

C: media de todos los ETF de la base de datos.

➤ **Sistemas basados en el contenido (Content-Based):**

▪ **Atributed-Based**

La idea de estos sistemas consiste en modelar los productos en base a atributos relevantes. Identificar las preferencias de los usuarios en referencia a estos atributos y de esta forma poder ofrecerle los productos que posean las características que la persona valora.

Veamos dos ejemplos de este tipo:

1. **Sistema basado en la descripción:**

El modelo que vamos a emplear consiste en cuantificar la **similitud entre textos**. Para poder hacer eso tenemos que transformar los textos en cantidades matemáticas, por lo que transformaremos el texto en vectores. Los dos métodos más populares que existen para la vectorización son : **CountVectorizer** y **TF-IDFVectorizer**.

• **CountVectorizer:**

Se trata del método más sencillo. Consiste en crear un vector de vocabulario y expresar los diferentes textos como vectores en base a este vector de vocabulario común.

• **TF-IDFVectorizer (Term Frequency-Inverse Document Frequency):**

En el caso anterior se asignaba el mismo peso a todas las palabras. Con este sistema se tiene en consideración esta idea y **se asignan pesos a cada palabra** según la siguiente fórmula:

$$w_{i,j} = tf_{i,j} * \log\left(\frac{N}{df_i}\right)$$

Ecuación 4-2: Peso asignado a cada palabra.

w_{ij} = peso de la palabra i en el documento j .

df_i = número de documentos que contiene la palabra i

N = número de documentos.

De esta forma se **asigna mayor peso a una palabra de un documento si esta aparece con mayor frecuencia en dicho documento y está presente en menos documentos.**

Para determinar la **similitud de dos documentos**, es decir de los vectores asociados que hemos construido con alguno de los dos métodos mencionados arriba se emplea la **similitud coseno** y su expresión es la siguiente:

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| * \|\mathbf{y}\|}$$

Ecuación 4-3: Cálculo similitud coseno (Cosine similarity)

Puede tomar cualquier valor entre 1 y -1. Cuanto mayor sea el resultado mayor es la similitud entre los documentos x e y .

Si queremos tener en cuenta que además cada usuario tiene una escala diferente de valoración , es decir que hay usuarios más exigentes que otros, se puede emplear lo que se conoce como **Similitud coseno ajustada** cuya fórmula es la siguiente:

$$\text{CosSim}(x, y) = \frac{\sum_i ((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}}$$

Ecuación 4-4: Cálculo similitud coseno ajustada

Finalmente, como medida de similitud también existe lo que se conoce como **similitud de Pearson**, que se obtiene realizando un pequeño ajuste a la ecuación anterior y se calcula de la siguiente forma:

$$\text{CosSim}(x, y) = \frac{\sum_i ((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}}$$

Ecuación 4-5: Similitud de Pearson.

En este caso en vez de centrarnos en la diferencia entre las valoraciones de un usuario y la media de las valoraciones de los usuarios, nos centramos en la diferencia entre las puntuaciones y la media de las puntuaciones que todos los usuarios han dado a un determinado elemento.

Esto puede entenderse como una medida de similitud basada en cuanto diverge cada usuario respecto a la media del comportamiento de las personas. Por ejemplo, si hay algo que a la mayoría de gente les gusta mucho, aquellas personas a las que no les guste van a aparecer como bastante similares ya que comparten opiniones que salen de lo habitual.

Al entrar a la aplicación se le pedirá al usuario que responda algunas preguntas acerca de sus preferencias como se ha comentado con anterioridad. A partir de estas puntuaciones se tratará de inferir sus preferencias utilizando el sistema que se acaba de explicar.

Para ello lo que se hace es tomar los textos descriptivos de todos los ETFs de la base de datos y compararlos usando el método **TF-IDFVectorizer**.

Finalmente, aquellos que obtengan mayores valores de similitud con los ETFs que mejor puntuación han recibido en el cuestionario son los que se ofrecerán en orden descendiente al inversor como recomendación.

2. Sistema basado en la metadata:

En este caso se siguen los mismos pasos que para el sistema anterior, con la diferencia del tipo de información que empleamos.

En primer lugar, escogemos los datos que vamos a emplear, para nuestro caso sería:

- Título
- Categoría
- Sector
- Asset Class Style.

Posteriormente creamos la “Sopa de metadatos”. En este caso se emplea **CounVectorizer** en vez del **TF-IDFVectorizer** ya que en este caso no tiene sentido penalizar la frecuencia de aparición.

➤ Filtros Colaborativos:

Los filtros colaborativos se aprovechan del **poder de la comunidad** para ofrecer recomendaciones. Los filtros colaborativos tratan de resolver el problema de predicción. Es decir, tratan de rellenar la matriz de **i usuarios** y **j productos**, donde cada elemento ij representa la puntuación asignada por el usuario i al producto j .

Dentro de estos podemos distinguir dos tipos: **Basados en el usuario** (User-based) y **basados en los productos** (item-based).

El principal problema que sufren este tipo de filtros se conoce como “**Cold start**” y es la falta de datos con las que se cuenta al principio. Por lo que el sistema no es eficiente hasta que se posee una base de datos lo suficientemente grande.

▪ Basados en el usuario (User-based):

Emplea el comportamiento de otros usuarios para hacer recomendaciones. La idea principal detrás de estos sistemas es **encontrar a personas similares a dicho usuario** para ofrecerle las cosas que dichos usuarios con un perfil similar han encontrado interesantes.

Como hemos dicho el objetivo es rellenar la matriz 2D con los usuarios en un eje y los productos en el otro.

Para poder establecer el grado de similitud entre dos usuarios solo debemos mirar los productos que dichos usuarios tengan en común, es decir los productos que ambos hayan puntuado. Esto tiene un problema dado que nuestra matriz probablemente este principalmente vacía, muchos usuarios no tendrán ningún producto en común y si por ejemplo tiene solo uno en común y lo han valorado con la misma puntuación el modelo no dirá que son 100% similares, por lo que a veces **será necesario establecer un mínimo en el número de productos comunes** que debe haber entre dos usuarios. También lo que podemos hacer es **ponderar la similitud** en base a la cantidad de valoraciones que compartan.

El primer paso es calcular el grado de similitud entre cada usuario y los demás, después podemos escoger los top N de esa lista y hacer una media de la puntuación que dieron esos usuarios al resto de productos ponderándola según el grado de similitud entre ellos.

De esta forma podemos calcular la predicción de cada usuario para cada producto teniendo en cuenta el grado de similitud entre usuarios de la siguiente forma:

$$r_{u,m} = \frac{\sum_{u', u' \neq u} sim(u, u') \cdot r_{u',m}}{\sum_{u', u' \neq u} |sim(u, u')|}$$

Ecuación 4-6: Grado de similitud entre usuarios (Hands-On Recommendation Systems with Python).

Donde $r_{u,m}$ representa la puntuación otorgada por el usuario u al elemento m .

Como podemos ver como medida de la similitud entre usuarios se ha empleado la similitud coseno.

Por otro lado, como se ha mencionado anteriormente existe otro problema que deberíamos tener en consideración y es el hecho de que las personas **puntúan en base a escalas diferentes** según las personas sean más optimistas y exigentes. Para solucionar este problema lo que hacemos es **normalizar** las escalas de cada usuario.

$$s(u, i) = \bar{r}_u + \frac{\sum_{v \in V} (r_{vi} - \bar{r}_v) * w_{uv}}{\sum_{v \in V} w_{uv}}$$

Ecuación 4-7: Similitud entre usuarios normalizada

Para el peso de la similitud podemos usar tanto la **similitud coseno** como la **correlación de Pearson** entre usuarios.

Añadido a esto podemos recurrir a lo que se conoce como “Demographis” que consiste en hacer lo mismo solo que esta vez reduciendo el universo de usuarios con los que comparamos

atendiendo a parámetros como pueden ser a edad, el sexo, la localización de una persona. Es decir, no tenemos en cuenta las valoraciones de todos los usuarios sino solo aquellos que **encajan en una cierta demografía.**

Otra opción sería también limitar el universo de usuarios a aquellos que cumplan con un **mínimo en la similitud.**

- **Basados en el producto (Item-based):**

En este caso en lugar de mirar a otros usuarios similares a ti y recomendarte cosas que les gustaron, miran las **cosas que le han gustado al usuario y le recomiendan cosas similares.**

Una ventaja que tiene este sistema frente al anterior es que normalmente tenemos más usuarios que productos por lo que el **cálculo de similitud será mucho menor** y velocidad de computación será mayor.

Además, normalmente los diferentes productos tendrán mayor número de valoraciones, que número de valoraciones hechas tendrán los usuarios por lo que **disminuye el problema de los vacíos en la matriz.**

Finalmente, la otra ventaja que tienen estos frente a los user-based es que la **similitud producto-producto es mucho más estable** que la similitud entre diferentes usuarios.

En este caso pensamos en los productos como un espacio donde cada usuario es una dimensión y hallamos la similitud entre los vectores de productos. Pero el procedimiento es el mismo que para el caso anterior.

- **Model-Based:**

Las técnicas que hemos visto hasta ahora de filtros colaborativos se conocen también como “memory-based”, ya que hacen solo uso de métricas de similitud y no hacen ningún uso de técnicas de machine learning.

Los conocidos como “**model-based**” si hacen uso de técnicas de machine learning para extraer predicciones.

1. Clustering:

Este método sería similar al filtro demográfico que acabamos de explicar solo que esta vez se tiene en consideración a aquellos **usuarios que pertenecen al mismo culster**. Para ello se utiliza el algoritmo K- means, KNN.

1. Identificamos los k usuarios vecinos más cercanos
2. Devolvemos la media de dichos usuarios para un determinado producto.

2. Matrix-Factorization:

Estos algoritmos consiguen encontrar características más amplias de usuarios y productos por si solos.

Aunque las matemáticas implicadas no nos dicen cómo llamar a estas características, son descritas por matrices que describen los atributos que se extraen de los datos.

- PCA:

Aplicando la técnica de PCA (explicada en capítulo x) a la matriz de productos- usuarios obtenemos una matriz con menor número de dimensiones, que constituyen las **características latentes** que se extraen de los datos. Como hemos dicho el **PCA no sabe cómo se llaman o que representan**, pero las encuentra.

En las siguientes imágenes se ha realizado un ejemplo de cómo sería su funcionamiento para un caso como el nuestro:

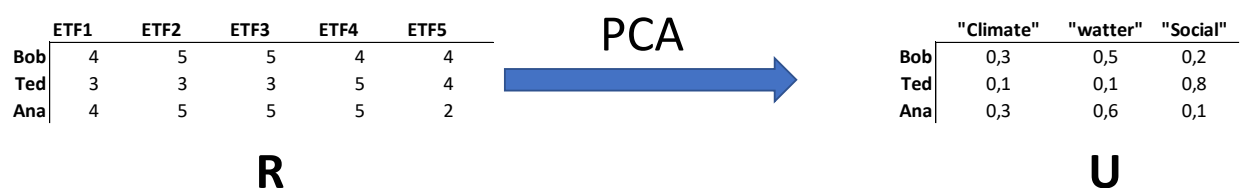


Figure 4-15: Aplicación PCA a la Matriz usuario-producto.

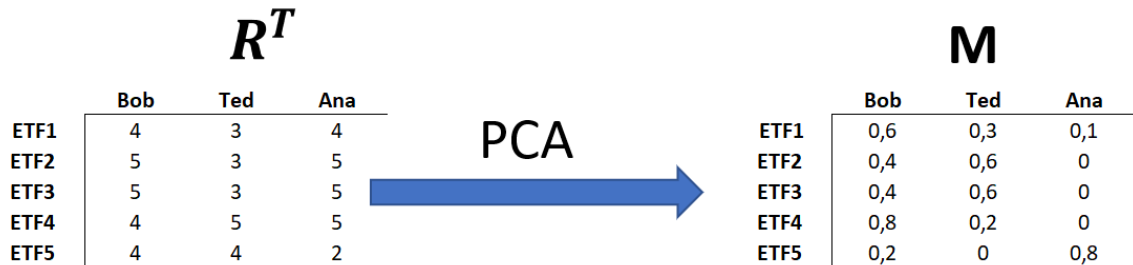
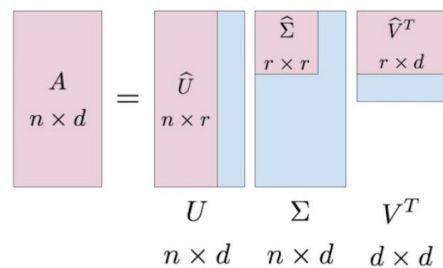


Figure 4-16: Aplicación PCA a la transpuesta de la matriz usuario producto.

Vemos como aplicando PCA a la matriz R y a su transpuesta podemos obtener las matrices U y M. A partir de las cuales se puede reconstruir la matriz R de la siguiente forma:

$$R = M \Sigma U^T$$

La matriz del medio es la matriz sigma que no es más que una matriz diagonal que sirve para escalar los valores.



$$\begin{matrix}
 \boxed{A} \\
 n \times d
 \end{matrix}
 =
 \begin{matrix}
 \boxed{U} \\
 n \times r
 \end{matrix}
 \begin{matrix}
 \boxed{\hat{\Sigma}} \\
 r \times r
 \end{matrix}
 \begin{matrix}
 \boxed{V^T} \\
 r \times d
 \end{matrix}$$

$U \quad \Sigma \quad V^T$
 $n \times d \quad n \times d \quad d \times d$

Figure 4-17: PCA (Libro: Hands on recommendation systems with Python)

- **SVD:**

Las iniciales SVD vienen de “**Singular value decomposition**” y es otro método para obtener la descomposición de la matriz R. Siguiendo el mismo principio que acabamos de ver.

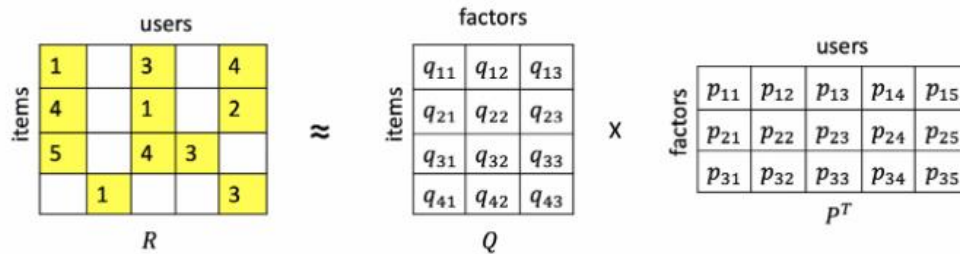


Figure 4-18: Representación de las puntuaciones como producto matricial.

$$A = U\Sigma V^T$$

$$A = R, \quad Q = U, \quad P^T = \Sigma V^T$$

Ecuación 4-8: Ecuación para SVD.

Como se ha podido deducir, existe un problema importante en estos desarrollos y es que necesitamos completar de alguna forma los valores que faltan de la matriz para poder aplicar eso. Por lo que se puede recurrir a rellenar los huecos con ceros o con valores medios. Se puede ver claramente que esto nos es la mejor opción por lo que han surgido nuevas técnicas para tratar de rellenar esta matriz.

Una de ellas es lo que se conoce como “**Stochastic Gradient Descent**” que básicamente intenta completar la matriz con los valores que minimicen el error.

No se entrará en el detalle matemático de este proceso, pero a grandes rasgos este es el procedimiento de este método:

1. Rellenamos los huecos de la matriz ya sea con ceros o valores aleatorios.
2. Tratamos de hacer la predicción.
3. Se calcula el error y se recalculan los valores de la matriz para reducir el error.
4. Se repite el proceso hasta que el sistema converge en la solución con mínimo error.

- **Deep Learning:**

Con el objetivo de completar los valores de esta matriz están surgiendo nuevas técnicas como la utilización de redes neuronales. Las redes neuronales que se emplean para estos sistemas de recomendación reciben el nombre de “**Restricted Boltzman Machines**”.

Pero la ciencia detrás de estas queda fuera de los objetivos de este proyecto.

- **Sistemas de recomendación Híbridos:**

Son aquellos que mezclan varios tipos de modelos de recomendación. Dado que como se ha visto cada modelo tiene sus propias ventajas y desventajas, con un modelo híbrido se pueden contrarrestar las desventajas de cada modelo.

Pongamos el ejemplo de Netflix que soluciona el problema del Cold Start de los filtros colaborativos usando un sistema basado en contenido y a medida que va obteniendo más datos introduce el filtro colaborativo.

4.4.3 ALGUNAS MÉTRICAS DE EVALUACIÓN

Existen numerosas formas de evaluar los sistemas de recomendación aquí solo se presentarán algunas de ellas que permiten cuantificar el desempeño de estos sistemas.

- **Accuracy:** Es la métrica más usada para evaluar el rendimiento de los modelos de clasificación y su fórmula es la siguiente:

$$Accuracy = \frac{\text{Número de predicciones correctas}}{\text{Total de predicciones}}$$

Ecuación 4-9: Ecuación Accuracy

- **Root mean square error (RMSE):** Una métrica muy usada para medir el desempeño de regresiones.

$$RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2}$$

Ecuación 4-10: Ecuación del RMSE

- **Métricas para clasificaciones binarias:**

- **TP:** True positive, se refiere a todos aquellos casos donde el valor real y el predicho son positivos.
- **TN:** True negative, se refiere a todos aquellos casos donde el valor real y el predicho son negativos.
- **FP:** False positive, se refiere a aquellos casos donde el valor real es negativo y el predicho positivo.
- **FN:** False negative, se refiere a aquellos casos donde el valor real es positivo y el predicho es negativo.

- **Precision:** Se define como el ratio entre el número de casos positivos correctos y el total de casos que fueron asignados como positivos.

$$Precision = \frac{TP}{TP + FP}$$

Ecuación 4-11: Ecuación Precision

- **Recall:** Ratio entre el número de casos identificados como positivos y el total de casos positivos en la base de datos.

$$Recall = \frac{TP}{TP + FN}$$

Ecuación 4-12: Ecuación Recall

- **F1 Score:** Es métrica que representa un balance entre *Precision* y *Recall*

$$F1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall}$$

Ecuación 4-13: Ecuación F1 Score

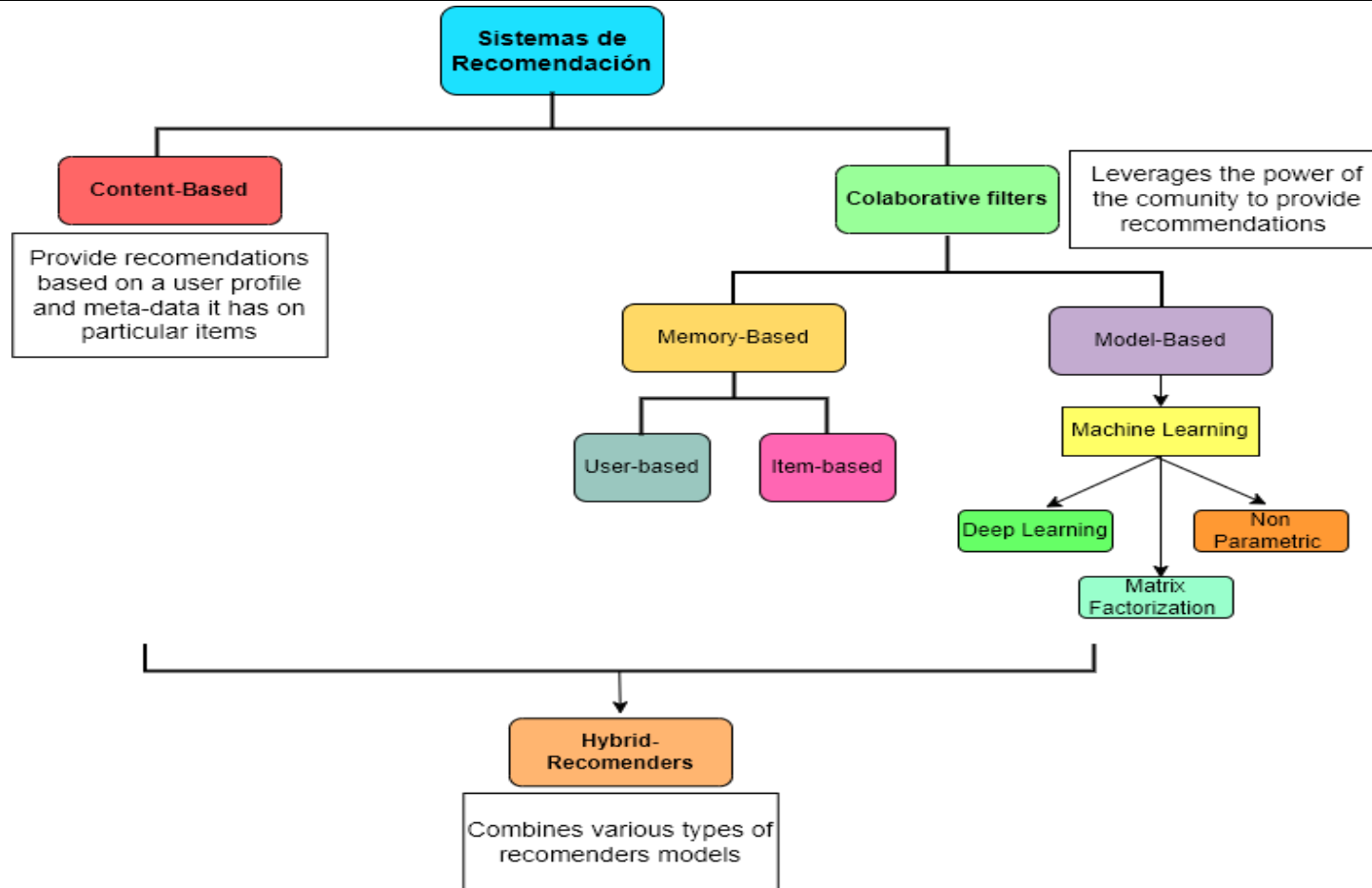


Figure 4-19: Esquema sistemas de recomendación.

4.4.4 DATOS

Uno de los mayores problemas que se han presentado a la hora de desarrollar la aplicación ha sido la obtención de datos. Por un lado, no ha sido posible encontrar una base de datos que contuviese las características ESG de los ETFs que no permitieran afinar mejor las recomendaciones.

Finalmente se ha creado una base de datos manualmente seleccionando alrededor de 25 ETFs diferentes con los datos obtenidos de la página **ETFDatabase**.

Como se vio en el punto 3.2.1 de inversión socialmente responsable existen diferentes aproximaciones, para este caso se ha empleado el enfoque de inversión con temática en sostenibilidad.

La información con las características de estos ETFs se almacenará en un Excel al que se accederá con el código y que contiene la siguiente información:

ETFname	Name	Descrip	Asset Class	Asset Class Size	Strategy	Region	ESG Score	ore Global Percent	Revenue Exposure to Environmental Impact (%)	Revenue Exposure to Social Impact (%)	Severe Governance Controversies (%)
TAN	Invesco S&P S&P 500 Index	The cleverly-named TAN d...	Equity	Multi-Cap	Fundamental	Developed Markets	7.7200	0.8803	0.6193	0.0000	0.0000
QCLN	First Trust NASDAQ Clean...	QCLN is a unique member...	Equity	Multi-Cap	Vanilla	North america	7.1400	0.5274	0.4080	0.0000	0.0000
XLK	Technology Select Sector...	State Street's XLK grants i...	Equity	Large-cap	Vanilla	North america	10.0000	1.0000	0.0875	0.0019	0.0002
NULG	Nuveen ESG Large-Cap Gr...	NULG invests in companie...	Equity	Large-cap	Growth	North america	8.4500	0.8345	0.0884	0.0612	0.0256
AQWA	Global X Clean Water ETF	AQWA provides exposure t...	Equity	<NA>	ESG	Global	10.0000	1.0000	0.2367	0.0211	0.0000
CGW	Invesco S&P Global Water...	This ETF offers exposure t...	Equity	Multi-Cap	Equal	Developed Markets	8.8800	0.8963	0.2080	0.0208	0.0000
EBLU	Ecofin Global Water ESG F...	EBLU offers exposure to de...	Equity	Multi-Cap	ESG	Developed Markets	9.7000	0.9705	0.1831	0.0351	0.0000
PHO	Invesco Water Resources...	This ETF offers exposure t...	Equity	Multi-Cap	Vanilla	North america	10.0000	1.0000	0.1462	0.0224	0.0000
PIO	Invesco Global Water ETF	This ETF offers exposure t...	Equity	Multi-Cap	Vanilla	Developed Markets	9.7300	0.9717	0.1355	0.0652	0.0000
OCEN	IQ Clean Oceans ETF	OCEN is passively manage...	Equity	<NA>	ESG	<NA>	8.9500	0.9056	0.1279	0.0495	0.0885
SDG	iShares MSCI Global Sust...	SDG tracks an ESG-facuse...	Equity	Large-cap	ESG	Developed Markets	8.0300	0.7516	0.4039	0.3372	0.0000

Tabla 4-1: Tabla características de los ETFs

4.4.5 MODELOS DE RECOMENDACIÓN EMPLEADOS EN EL PROTOTIPO

Una vez hemos visto los diferentes sistemas de recomendación que existen toca decidir cual o cuales de ellos serán los que se empleen para el prototipo que se desea construir.

Como ya se mencionó el sistema de recomendación del Robo-Advisor que se desea crear debe recomendar diferentes fondos o ETF's de materia ESG a los clientes.

De los sistemas que hemos vistos se han creado los siguientes:

- **Top 5** de los ETFs más seleccionados por los usuarios.
- Sistema **basado en el contenido** que emplea las descripciones de los ETFs para evaluar su similitud.
- Sistema **basado en el contenido** que emplea otras **características** de los ETFs para evaluar su similitud.
- Recomendación que despliega los ETFs con mejores predicciones de puntuación basándose en la media ponderada según la similitud con otros usuarios. (**Colaborative filter – Memory based – User-based**).
- **Sistema Híbrido** que combina un sistema basado en contenido según la similitud entre ETFs a partir de sus descripciones y un sistema que emplea SVD.
- **Sistema Híbrido** que combina un sistema basado en contenido según la similitud entre ETFs a partir de sus descripciones y un sistema que emplea el (**Colaborative filter – Memory based – User-based**) .

Capítulo 5. IMPLEMENTACIÓN PRÁCTICA: MODELO

ROBO-ADVISOR.

En este capítulo se presentará todo el proceso de construcción del prototipo de la plataforma web para el asesoramiento automatizado que implementará varios de los algoritmos que se han expuesto en capítulos anteriores.

5.1 DISEÑO INTERFAZ

Todo el código del modelo ha sido realizado en lenguaje **Python** y para la implementación web de este se ha optado por usar la librería **Streamlit**.

Estructura del código:

En la imagen inferior se presenta un esquema de la estructura que tiene aplicación web que se ha construido a modo de prototipo:

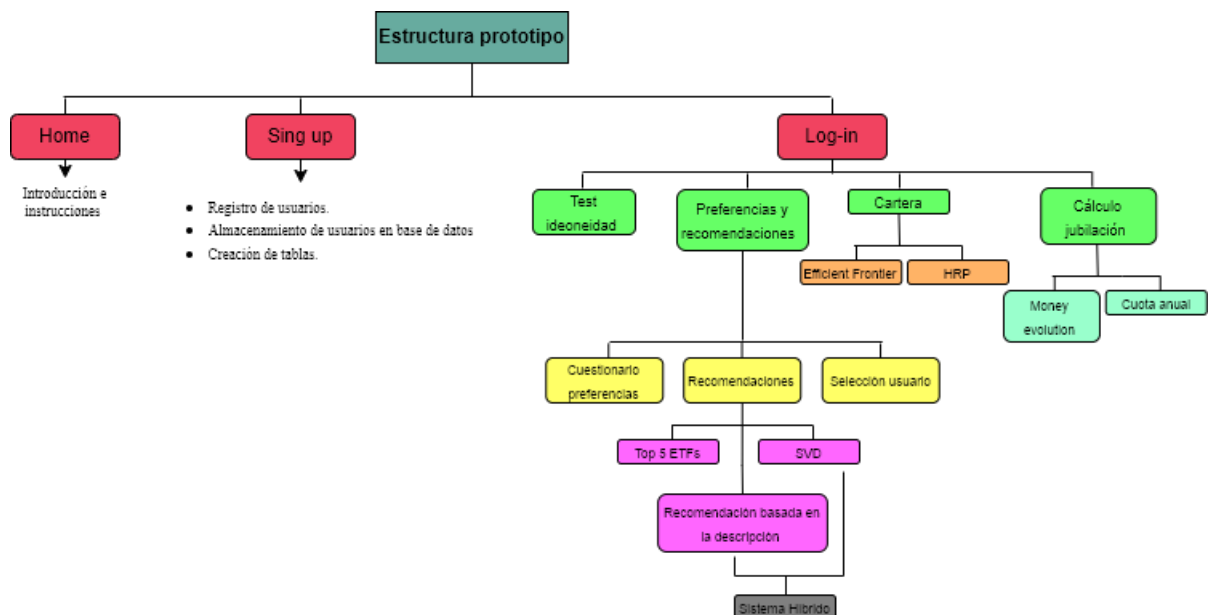


Figure 5-1: Esquema de la estructura de la aplicación Web

En primer lugar, cada persona que ingrese a la web deberá crear un nombre de usuario y contraseña. Una vez hecho esto podrá acceder a la aplicación que cuenta con tres pestañas diferentes:

1. **Test de Idoneidad:** Es aquí donde cada nuevo usuario debe responder a las preguntas del test de idoneidad, sin embargo, este no se usará para asignar un perfil de riesgo, sino que será meramente ilustrativo.

A todos los usuarios se les asignará la cartera más óptima sin atender al perfil de riesgo, con el fin de simplificar la implementación del prototipo.

2. **Cuestionario ESG:** Una vez finalizado el test de idoneidad, cada usuario deberá responder y ofrecer una serie de valoraciones de varios ETFs. que servirá como referencia para el sistema de recomendación.

Dentro de este apartado encontramos 3 pasos:

1. Cuestionario de preferencias: Donde a cada usuario se le presentan 5 ETFs aleatorios para que los valore con una puntuación entre 1y 5. Con el fin de entender sus gustos.
2. Una vez hecho esto al usuario se le ofrecen las recomendaciones a partir de los sistemas de recomendación implementados.
3. Finalmente, el usuario selecciona los 5 ETFs que quiere que formen parte de su cartera. Que pueden ser o no los que se han ofrecido a modo de recomendación por el sistema.

3. **Portfolio:** Con los 5 ETFs seleccionados se procede a la selección de los pesos que estos tendrán dentro del portafolio del inversor.

Dentro de este apartado el sistema cuenta con dos opciones:

- Optimización: En este caso se ofrecen los resultados de la colocación de activos utilizando las técnicas de MVP y HRP explicados anteriormente.
- Backtest: Se hace un back test de ambas técnicas para observar las diferencias y la evolución que hubiesen tenido ambas carteras en el pasado.

4. Cálculo Jubilación: Esta parte se ha añadido como extra y no tiene relación con las partes anteriores. Consiste en ofrecer al usuario la posibilidad de simular diferentes escenarios relacionados con su patrimonio futuro:

Dentro de este apartado se ofrecen 2 variantes:

- Evolución del dinero: El usuario introduce las variables de inversión, rentabilidad, inflación, edad de jubilación etc. y el sistema le devuelve una gráfica con la evolución de su patrimonio según los diferentes escenarios.
- Cuota anual: En este caso se busca que el usuario establezca un objetivo y se la muestra la cantidad de dinero anual que debería invertir según el escenario establecido para cumplir con dicho objetivo.

5.2 CÓDIGO BASE

A continuación, se presentan fragmentos del código principal que se ejecuta para construir la aplicación web.

```
def main():
    '''Simple login App'''

    st.title("Log in App")

    choice = st.sidebar.selectbox("Menu", menu_list)

    if choice == menu_options['home']:
        home_choice()

    elif choice == menu_options['login']:
        login_choice()

    elif choice == menu_options['signup']:
        signup_choice()

if __name__ == '__main__':
    main()
```

Código **main**
del programa.

Código 5-1: Función principal (main)

```
def home_choice():
    st.subheader("Home")

def login_choice():
    st.subheader("Login Section")
    username = st.sidebar.text_input("User Name")
    password = st.sidebar.text_input("Password", type='password')
    if st.sidebar.checkbox("Login"):
        hashed_pswd = make_hashes(password)
        result = sqt.login_user(username, check_hashes(password, hashed_pswd))
        if result:
            login_result(username)
        else:
            st.warning("Incorrect Username/password")

def signup_choice():
    st.subheader("Create a New Account")
    new_user = st.text_input('Username')
    new_password = st.text_input("Password", type='password')
    if st.button("SignUp"):
        sqt.create_userstable()
        sqt.create_userprofiletable()
        sqt.create_userspuntuations()
        sqt.create_usersselection()
        sqt.add_userdata(new_user, make_hashes(new_password))
        st.success(" you have successfully created a valid Account")
        st.info("Go to login")
```

Verificación de nombre y contraseña del usuario para permitir acceso.

Alta de usuario en la plataforma y creación de tablas.

Código 5-2: Opciones del main.

```
def make_hashes(password):
    return hashlib.sha256(str.encode(password)).hexdigest()

def check_hashes(password, hashed_text):
    if make_hashes(password) == hashed_text:
        return hashed_text
    return False
```

Encriptación de la contraseña.

Función para comprobar que la contraseña introducida coincide con la correcta.

Código 5-3: Funciones para la encriptación de contraseñas.

```
def task_questionnaire(username):
    st.subheader(task_options['questionnaire'])
    df = cu.user_input_features()
    st.dataframe(df)
    if st.checkbox("Cuestionario terminado"):
        sqt.add_userprofile(username, df)
        #user_answers = sqt.view_users_answers()

def task_preferences(username):
    st.title("Preferencias")
    st.subheader('Listado ETFs disponibles')
    cu.RandomETFpuntuation(username) #actualiza tabla con las puntuaciones de los usuarios por cada ETF
    selected ETFs = cu.Seleccion_final(username)
    if st.checkbox("Selección terminada"):
        # creamos base de datos con los ETFs que selecciona cada usuario
        sqt.add_usersselection(username, selected ETFs)
```

Función para desplegar el test de idoneidad.

Función para desplegar la página de preferencias

Código 5-4

```
def task_wallet(username):  
    # función para extraer la información de los ETFs seleccionados por el usuario  
    st.header("Mi Cartera")  
    selected ETFs = sqt.check_userselection(username)  
    opciones = ['Optimización', 'Backtest']  
    choice_carteras = st.selectbox("Elige una opción: ", opciones)  
  
    if choice_carteras == 'Optimización':  
        st.subheader('Cálculo cátera óptima random allocation')  
        eft.RandomAllo(selected ETFs)  
        st.subheader('optimizacion efficientet')  
        eft.opt_eff(selected ETFs)  
        st.subheader('Cálculo con libreria')  
        opt.HRP_Eff(selected ETFs)  
    else:  
        st.subheader('Backtest de la optimización de la cartera con los métodos MVP y HRP')  
        hrp.EjecucionHRP(selected ETFs)  
  
def task_retirement():  
    opciones = ['Calculo 1', 'Cálculo 2']  
    choice_jubilacion = st.selectbox("Calculo Jubilación: ", opciones)  
  
    if choice_jubilacion == 'Calculo 1':  
        jl.jubilacion()  
  
    elif choice_jubilacion == 'Cálculo 2':  
        jl.jubilacion2()  
  
def login_result(username):  
    st.success("Logged In as {}".format(username))  
    task = st.selectbox("Task", task_list)  
  
    if task == task_options['questionnaire']:  
        task_questionnaire(username)  
  
    elif task == task_options['preferences']:  
        task_preferences(username)  
  
    elif (task == task_options['wallet']) :  
        task_wallet(username)  
  
    elif task == task_options['retirement']:  
        task_retirement()
```

Función para desplegar la página de la cartera del usuario.

Función para desplegar la página para la simulación de la jubilación.

Función para redirigir al usuario a cada una de las diferentes páginas dependiendo de su selección.

Código 5-5

5.3 CÓDIGO SQL

Para la gestión de la información de todos los usuarios es necesario emplear una base de datos externa. Para ello Emplearemos **SQL** y para su manejo desde Python se emplea el módulo **MySQL**.

La estructura de las diferentes tablas que compondrán la base de datos y que se estarán manejando durante todo el programa son las siguientes:

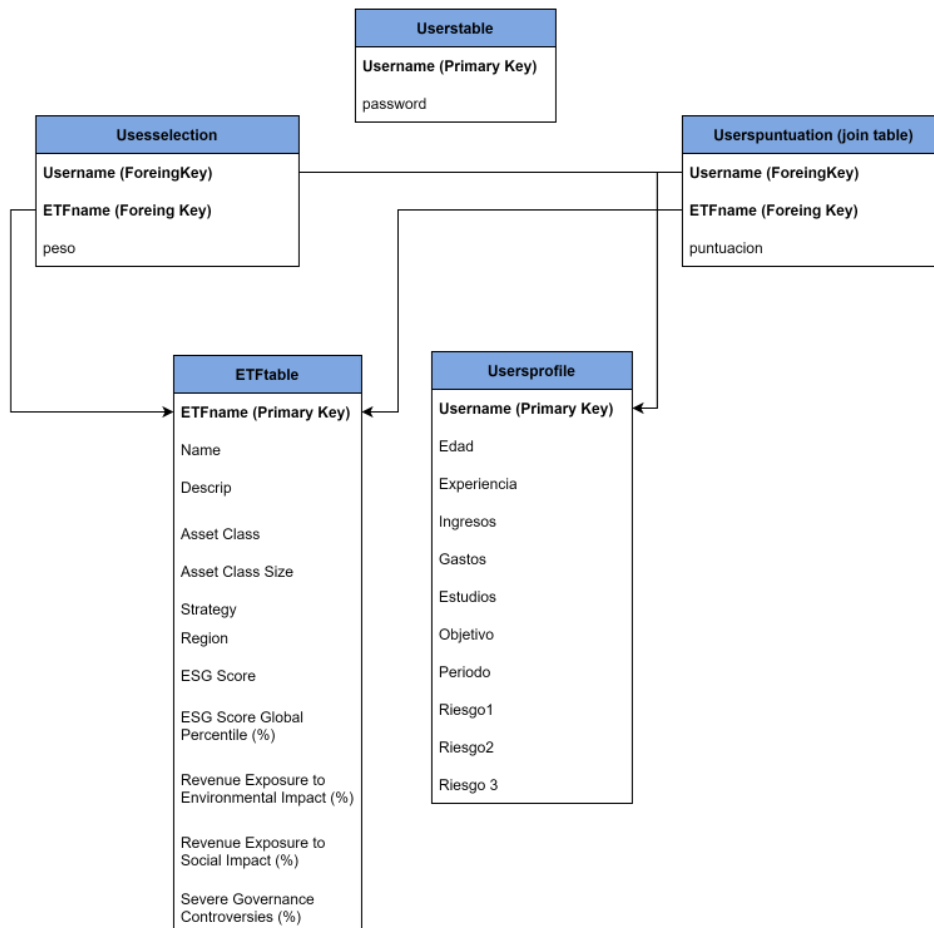


Figure 5-2: Estructura base de datos (Tablas)

Para la gestión de las ordenes relacionadas con la base de datos se ha creado un script de Python llamado **Mysql** y que se importará desde el Script principal para poder acceder a sus funciones.

En las siguientes imágenes se presentan fragmentos de este código:

```
import mysql.connector
import streamlit as st
from pandas import DataFrame
import itertools

connection = mysql.connector.connect(
    host = "localhost",
    user = "root",
    passwd = "Mysql1201606556",
    database = "tfm"
)

cursor = connection.cursor()

def execute_query(sql_query):
    cursor.execute(sql_query)
    connection.commit()
```

Importación de librerías

Establecimiento de la conexión con la base de datos.

Función para ejecutar queries.

Código 5-7: Establecimiento de la conexión con la base de datos.

```
def login_user(username, password):
    cursor.execute('SELECT * FROM userstable WHERE username = %s AND password = %s', (username, password))
    data = cursor.fetchall()
    return data

#Create Userstable
def create_userstable():

    query_create_userstable = '''
    CREATE TABLE IF NOT EXISTS userstable (
        username VARCHAR(200),
        PRIMARY KEY (username),
        password TEXT
    );
    '''

    execute_query(query_create_userstable)

def add_userdata(username, password):
    cursor.execute('INSERT IGNORE INTO userstable(username,password) VALUES (%s,%s)',(username, password))
    connection.commit()
```

Introducir nombre y contraseña encriptada en la tabla de usuarios.

Creación tabla userstable.

Actualizar userstable.

Código 5-6: Funciones para la creación y manipulación de tablas(userstable).

```
# Create userspuntuacionstable
def create_userspuntuations():
    query_create = '''
    CREATE TABLE IF NOT EXISTS userspuntuations(
        username VARCHAR(200),
        ETFname VARCHAR(200),
        puntuacion INT
        #FOREIGN KEY (username) REFERENCES usersprofile(username),
        #FOREIGN KEY (ETFname) REFERENCES ETFtable(ETFname)
    );
    '''

    execute_query(query_create)

# Add userspuntuaciondata
def add_userspuntuations(username,dictionary):
    cursor.execute('DELETE FROM userspuntuations WHERE username = %s', (username,))
    for ETF, punt in dictionary.items():
        #st.write(username,ETF,punt)
        cursor.execute('INSERT INTO userspuntuations(username,ETFname,puntuacion) VALUES (%s,%s,%s)', (username,ETF,punt))
    connection.commit()
```

Crear userspuntuations table.

Actualizar userspuntuations.

Código 5-8: Funciones para la creación y manipulación de tablas(userspuntuacion).

```
# crear usuarselectiontable
def create_usersselection():
    query_create = '''
        CREATE TABLE IF NOT EXISTS usersselection(
            username VARCHAR(200),
            ETFname VARCHAR(200)
            #FOREIGN KEY (username) REFERENCES usersprofile(username),
            #FOREIGN KEY (ETFname) REFERENCES ETFtable(ETFname)
        );
    '''
    execute_query(query_create)
# actualizar tabla nexa
def add_usersselection(username,selection_list):
    #Se eliminan registros anteriores del usuario
    cursor.execute('DELETE FROM usersselection WHERE username = %s',(username,))
    #Se insertan los nuevos valores
    for ETFname in selection_list:
        st.write(ETFname)
        cursor.execute('INSERT INTO usersselection(username,ETFname) VALUES (%s,%s)',(username,ETFname))
        connection.commit()

def check_userselection(username):
    cursor.execute('SELECT ETFname FROM usersselection WHERE username = %s',(username,))
    resultados = cursor.fetchall()#devuelve una lista con los etfs del usuario
    connection.commit()
    return resultados
```

Funciones referentes a la tabla *usersselectio*.

Código 5-9: Funciones para la creación y manipulación de tablas(*usersselection*).

```
def importuserspunct():
    cursor.execute('SELECT * FROM userspuntuations')
    records = cursor.fetchall()
    connection.commit()
    df = DataFrame(records)
    return df

def importusersprofile():
    cursor.execute('SELECT * FROM usersprofile')
    records = cursor.fetchall()
    connection.commit()
    df = DataFrame(records)
    return df
```

Transformar la tabla *userspuntuations* en un DataFrame

Transformar la tabla *usersprofile* en un DataFrame

Código 5-10: Funciones para transformar tabla SQL a DataFrame


```
# User Item matrix creation
def CreateUserMatrix():

    cursor.execute('SELECT DISTINCT username from userspuntuations')
    usuarios = cursor.fetchall() # nos devuelve una lista de listas
    flat_list = itertools.chain(*usuarios)
    usuarios = List(flat_list)
    cursor.execute('SELECT DISTINCT ETFname from userspuntuations')
    ETFs = cursor.fetchall()
    flat_listETF = itertools.chain(*ETFs)
    ETFs = List(flat_listETF)

    cursor.execute('SELECT * FROM userspuntuations')
    records = cursor.fetchall()
    connection.commit()

    matrix = DataFrame(columns = ETFs, index = usuarios)
    for row in records:
        matrix.loc[row[0],row[1]]= row[2]

    #st.write(matrix)
    return matrix
```

Función que crea la matriz de con las puntuaciones de los usuarios a los diferentes ETFs, que se usará para hacer SVD.

Código 5-11

```
def user_input_features():

    Edad = st.text_input('1. Introduce tu edad ',20)
    Genero = st.selectbox('2. Indica tu género', Genero_List)
    Experiencia = st.selectbox(
        '3. En cuanto a la inversión en Fondos, ETFs o acciones individuales me describiría a mi mismo como.. ', Experiencia_List)
    Ingresos = st.selectbox(
        '4. Mis actuales y futuras fuentes de ingresos son ... ', Ingresos_List)
    Gastos = st.selectbox(
        '5. ¿Qué porcentaje de tus ingresos se van en gastos? ', Gastos_List)
    Estudios = st.selectbox(
        '6. ¿ Cuál es su nivel de estudios? ', Estudios_List)
    Objetivo = st.selectbox(
        '7.¿Qué rentabilidad buscas para tu cartera de inversión? ', Metas_List)
    Periodo = st.selectbox(
        '8. Planeo empezar a recoger el dinero de mis inversiones en... ', Periodo_List)
    Riesgo = st.selectbox(
        '9. Imagina que su cartera de inversiones pierde un 10% de su valor en un mes ¿Qué harías? ', Riesgo1_List)
    Riesgo2 = st.selectbox(
        '10. Durante las caídas del mercado tiendo a vender partes de mis activos más arriesgados e invertir en activos más seguros. ', Riesgo2_List)
    Riesgo3 = st.selectbox(
        '11.Si piensas en la palabra "Riesgo" ¿cuál de las siguientes palabras te viene a la mente? ', Riesgo3_List)

    data = {'Edad': Edad,
            'Género': Genero,
            'Experiencia': Experiencia,
            'Ingresos': Ingresos,
            'Gastos': Gastos,
            'Estudios': Estudios,
            'Objetivo': Objetivo,
            'Periodo': Periodo,
            'Riesgo1': Riesgo,
            'Riesgo2': Riesgo2,
            'Riesgo3': Riesgo3}

    features = pd.DataFrame(data, index=[0])
    return features
```

Código 5-12: Función para obtener respuestas del test de idoneidad del usuario

```
def RandomETFpuntuacion(username):
    ETFs = read_excel('ETFs(excel).xlsx')
    ETFs.set_index('ETFname', inplace=True)
    st.write(ETFs)
    #st.dataframe(ETF)
    st.title('¿Cuales son tus preferencias?')
    st.write("""
    Puntua los siguientes ETFs del 1 al 5
    """)
    #Selección de 5 EFTs aleatorios para valorar por cada usuario
    #columns_names = ETF.index.values
    names_list = List(ETFs.index.values)
    lista = sample(names_list,5)

    Nom ETF1 = lista[0]
    Nom ETF2= lista[1]
    Nom ETF3= lista[2]
    Nom ETF4 = lista[3]
    Nom ETF5= lista[4]
    Des ETF1 = ETFs['Descrip'][Nom ETF1]
    Des ETF2 = ETFs['Descrip'][Nom ETF2]
    Des ETF3 = ETFs['Descrip'][Nom ETF3]
    Des ETF4 = ETFs['Descrip'][Nom ETF4]
    Des ETF5 = ETFs['Descrip'][Nom ETF5]

    st.write(Nom ETF1)
    st.write(Des ETF1)
    punt ETF1 = st.text_input('Puntua este ETF entre 1 y 5 :',3,key=0)
    int(punt ETF1)
    st.write(Nom ETF2)
    st.write(Des ETF2)
    punt ETF2 = st.text_input('Puntua este ETF entre 1 y 5 :',3,key=1)
    st.write(Nom ETF3)
    st.write(Des ETF3)
    punt ETF3 = st.text_input('Puntua este ETF entre 1 y 5 :',3,key=2)
    st.write(Nom ETF4)
    st.write(Des ETF4)
    punt ETF4 = st.text_input('Puntua este ETF entre 1 y 5 :',3,key=3)
    st.write(Nom ETF5)
    st.write(Des ETF5)
    punt ETF5 = st.text_input('Puntua este ETF entre 1 y 5 :',3,key=4)

    st.write(type(punt ETF1))
```

Descargamos información

Se seleccionan 5 ETFs de forma aleatoria.

Extraemos valoraciones del usuario.

Código 5-13:Primera parte de la función que obtiene las puntuaciones de los ETFs por parte del usuario

```
puntuaciones= {
    lista[0]:punt ETF1,
    lista[1]: punt ETF2,
    lista[2]: punt ETF3,
    lista[3]: punt ETF4,
    lista[4]: punt ETF5}

if st.checkbox("Selección terminada",key=1):
    add_userspuntuations(username,puntuaciones)
    st.subheader('El Top 5 de ETFs seleccionados por los usuarios son: ''')
    st.write('Top 5 seleccionados')
    sqt.Top ETFs()
    matrix = sqt.CreateUserMatrix()
    st.write('evaluation SVD')
    evalu()
    st.write('Hybrid SVD')
    #Mostramos otras recomendaciones hechas por el sistema hibrido de recomendaci
    hy.hybrid_recom(username)
    st.write('Hybrid simple')
    hy.hybrid(username)
    #Puntuación del modelo

return puntuaciones
```

Se guardan resultados y se llaman a las funciones encargadas de ofrecer recomendaciones.

Código 5-14:Primera parte de la función que obtiene las puntuaciones de los ETFs por parte del usuario.

```
def Seleccion_final(username):
    ETFs = read_excel('ETFs(excel).xlsx')
    ETFs.set_index('ETFname', inplace=True)
    lista = list(ETFs.index.values)
    st.subheader('selección ETFs')
    User_final_selec = st.multiselect('Selecciona los 5 EFTs que quieres que formen parte de tu cartea', lista)
    return User_final_selec
```

Código 5-15: Función que recoge la selección final del usuario.

5.4 CÓDIGO DE RECOMENDACIONES

A continuación, se presenta el código de los diferentes algoritmos de recomendación mencionados anteriormente junto con una explicación más detallada.

5.4.1 TOP 5 SELECCIONADOS

En primer lugar, siguiendo el ejemplo de Netflix, se pretende ofrecer un listado de los Top 5 ETFs más elegidos. Este es el modelo de recomendación más simple, para el cual solo será necesario acceder a la base de datos donde se almacenan los ETFs escogidos por los inversores y extraer aquellos más repetidos en orden descendente. En caso de ser el primer usuario en acceder a la plataforma se despliega un mensaje indicando que no existen registros anteriores con los que ofrecer recomendaciones.

```
def Top ETFs():
    #En el caso de que sea el primer usuario no podremos ofrecer esta recomendacion, en este caso hacemos lo siguiente:
    cursor.execute('SELECT COUNT(*) FROM tfm.usersselection')
    numRegistros = cursor.fetchall()
    numRegistros = numRegistros[0][0]
    if numRegistros == 5:
        st.write('Es usted el primero en registrarse , aun no existe lista de favoritos')
        results = 0
    else:
        # selecciona la columna con los nombre de los ETF y cuenta las veces que aparece ordenandolas de mas a menos
        sql_query = 'SELECT ETFname, COUNT(*) AS numberofETF FROM tfm.usersselection GROUP BY ETFname ORDER BY COUNT(*) DESC'
        cursor.execute(sql_query)
        resultados = cursor.fetchall()
        connection.commit()
        results = DataFrame.from_dict(resultados).rename(columns = {0:'ETFname',1:'num_selec'})
        # ahora para visualizar los resultados:
        st.write(results.iloc[0:5,:])
    return results
```

A partir de la tabla *usersselection* ordenamos y filtramos aquellos que más a parecen y los desplegamos en orden descendente.

Código 5-16: Función Top 5 ETFs seleccionados.

5.4.2 SISTEMA DE RECOMENDACIÓN BASADO EN LA DESCRIPCIÓN

Este algoritmo se basa en la aplicación de TF-IDFVectorizer a las descripciones de los ETFs.

La función que se ha creado para ello recibe como input el nombre del ETF y devuelve los tres ETFs más similares según la descripción.

Este algoritmo no se implementará de forma aislada, sino que funcionará junto al SVD para crear el sistema híbrido que ofrezca las recomendaciones finales.

Nota: El código que se ha empleado para este sistema se ha obtenido a partir de la adaptación del código que emplea el método *TF-IDFVectorizer*, presentado en el libro “Hand son Recommender Systems with Python”.

```
#Import data de los ETFs
#Importamos los datos de los ETFs
ETFs = read_excel('ETFs(excel).xlsx', index_col=0)
ETFs.reset_index(inplace = True)

#En primer lugar necesitamos la matriz de ratings
#Import TfIdfVectorizer from the scikit-learn library
from sklearn.feature_extraction.text import TfidfVectorizer

#Define a TF-IDF Vectorizer Object. Remove all english stopwords
tfidf = TfidfVectorizer(stop_words='english')

#Replace NaN with an empty string
ETFs['Descrip'] = ETFs['Descrip'].fillna('')

#Construct the required TF-IDF matrix by applying the fit_transform method on the Description feature
tfidf_matrix = tfidf.fit_transform(ETFs['Descrip'])

# Import linear_kernel to compute the dot product
from sklearn.metrics.pairwise import linear_kernel

# Compute the cosine similarity matrix
cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)

#Construct a reverse mapping of indices and ETFs names, and drop duplicate titles, if any
indices = pd.Series(ETFs.index, index=ETFs['ETFname']).drop_duplicates()

# Function that takes in the ticker symbol as input and gives recommendations
def content_recommender(ticker, cosine_sim=cosine_sim, df=ETFs, indices=indices):
    idx = indices[ticker]
    # Get the pairwise similarity scores of all ETFs with that ETF
    # And convert it into a list of tuples as described above
    sim_scores = list(enumerate(cosine_sim[idx]))
    # Sort the ETFs based on the cosine similarity scores
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    # Get the scores of the 3 most similar movies. Ignore the first movie.
    sim_scores = sim_scores[1:4]
    # Get the ETFs indices
    movie_indices = [i[0] for i in sim_scores]
    # Return the top 10 most similar ETFs
    return df[['ETFname', 'Descrip']].iloc[movie_indices]
```

Función que devuelve nombre y descripción de los 3 ETFs más similares al que se introduce en la función., basándose en la

Código 5-17: Código del Sistema de recomendación basado en la descripción.

5.4.3 SISTEMA BASADO EN CONTENIDO A PARTIR DE LAS CARACTERÍSTICAS DE LOS ETFs

Las siguientes funciones se han obtenido adaptando el código de: ‘Metadata-based recommender’ del libro ‘*Hands-On Recommender systems with python*’.

Lo que hace es emplear el método de CountVectorizer al vector creado a partir de las características de **Asset Class, Asset Class Size, Strategy, Region** y por lo tanto las recomendaciones que se ofrecen no tiene por qué ser similares en cuanto a la temática se refiere.

```
def sanitize(x):
    if isinstance(x, list):
        return [str.lower(i.replace(' ', '')) for i in x]
    else:
        if isinstance(x, str):
            return str.lower(x.replace(' ', ''))
        else:
            return ''

def create_soup(x):
    return ' '+x['Asset Class']+' '+x['Asset Class Size']+' '+x['Strategy ']+ ' '+x['Region ' ]

def metadata_recommender(ticker):
    df = ETFs
    df
    features = ['Asset Class', 'Asset Class Size', 'Strategy ', 'Region ' ]
    #Aplicamos la función sanitize que elimina los espacios y convierte todo a minusculas
    for feature in features:
        df[feature] = df[feature].apply(sanitize)
    #Creamos una sopa con todas las características anteriores
    df['soup'] = df.apply(create_soup, axis=1)
    #Definimos un CountVectorizer object
    count = CountVectorizer(stop_words='english')
    #Creamos el vector a partir de la sopa
    count_matrix = count.fit_transform(df['soup'])
    #Aplicamos la similitud coseno a estos
    cosine_sim2 = cosine_similarity(count_matrix, count_matrix)

    df = df.reset_index()
    indices2 = pd.Series(df.index, index=df['ETFname'])
    results = content_recommender(ticker, cosine_sim2, df, indices2)
    st.write('Resultados:')
    st.write(results)
```

Función que elimina y pone todo en minúsculas.

Función que ofrece recomendaciones finales según los ETFs más similares según sus características al introducido en la función.

Código 5-18: Código del Sistema de recomendación basado en las características.

5.4.4 FILTRO COLABORATIVO SIMPLE BASADO EN EL USUARIO

En este caso se pretende mostrar el funcionamiento de un algoritmo colaborativo de los conocidos como “memory-based” donde solo se aplicará la **media de las valoraciones de otros usuarios similares** al usuario al que se pretende ofrecer las recomendaciones. De nuevo, estas funciones se han creado a partir de las propuestas en: del libro ‘*Hands-On Recommender systems with python*’.

Además, en este caso se dividen los datos en *train* y *test* set para poder evaluar posteriormente la eficacia del modelo.

```
# importamos la tabla de puntuaciones
ratings = sqt.importuserspunct()
ratings.columns=['username', 'ETFname', 'Puntuacion']

#Asignamos a la variable x la tabla de datos original y a la variable y la columna de usuario
x = ratings.copy()
y = ratings['username']

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size= 0.25 ,stratify = y, random_state = 42)
#Importamos los datos de los ETFs
ETFs = read_excel('ETFs(excel).xlsx', index_col=0)
#En primer lugar necesitamos la matriz de ratings
matrix = sqt.CreateUserMatrix()
# creamos una matrix que rellene con cero los valores nulos
matrix0= matrix.copy().fillna(0)
#calculamos la cosine similarity matrix usando la matriz anterior
cosine_sim = cosine_similarity(matrix0, matrix0)
#Convert into pandas dataframe
cosine_sim = DataFrame(cosine_sim, index=matrix.index, columns=matrix.index)
```

- Cargamos datos
- Dividimos en train y test sets.
- Importamos matriz calculamos la similitud coseno

Código 5-19: Preparación de los datos del modelo.

```
#User Based Collaborative Filter using Weighted Mean Ratings
def cf_user_wmean(username, ETF):
    #Check if movie_id exists in r_matrix
    if (username in matrix.index) and (ETF in matrix.columns):
        #Get the similarity scores for the user in question with every other user
        sim_scores = cosine_sim[username]
        #Get the user ratings for the ETF in question
        m_ratings = matrix[ETF]
        #Extract the indices containing NaN in the m_ratings series
        idx = m_ratings[m_ratings.isnull()].index
        #Drop the NaN values from the m_ratings Series
        m_ratings = m_ratings.dropna()
        #Drop the corresponding cosine scores from the sim_scores series
        sim_scores = sim_scores.drop(idx)
        #compute the final weighted mean
        wmean_rating = np.dot(sim_scores, m_ratings)/ sim_scores.sum()
    else:
        #Default to a rating of 3.0 in the absence of any information
        wmean_rating = 3.0
    return wmean_rating
```

Función que predice la puntuación que un usuario otorgaría a un determinado ETF según las valoraciones de usuarios similares.

Código 5-20:Filtro colaborativo según la similitud con otros usuarios.

```
#Funcion que devuelve el RMSE
def rmse(y_true, y_pred):
    return np.sqrt(mean_squared_error(y_true, y_pred))

# Función que nos devuelve el RSME del modelo que le enviamos
def score(cf_model):
    # creamos una lista de usuarios-ETFs
    id_pairs = zip(x_test['username'],x_test['ETFname'])

    # predecimos los valores de la variable de salida aplicando el modelo
    y_pred = np.array([cf_model(username, ETF) for (username, ETF) in id_pairs])

    # extraemos los valores reales con los que compararemos las predicciones
    y_true = np.array(x_test['Puntuacion'])

    #devolvemos el valor del RMSE
    return rmse(y_true,y_pred)
```

Funciones para medir los resultados del modelo.

Código 5-21: Funciones para evaluar eficacia del modelo.

5.4.5 SVD

Para este sistema se ha creado un archivo llamado **SVD.py** en el que se encuentran las siguientes funciones:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from surprise import Dataset
from surprise import Reader

import streamlit as st
import SQLdesde0 as sqt

#CREATE SURPRISE DATA SET
ratings = sqt.importuserspunct()
ratings.columns=['username','ETFname','Puntuacion']
reader = Reader(rating_scale=(1, 5))
data = Dataset.load_from_df(ratings, reader)

#TRAINING AND CROSS-VALIDATION SVD MODEL
from surprise import SVD
from surprise.model_selection import cross_validate

svd = SVD(verbose=True, n_epochs=10)
cross_validate(svd,data, measures=['RMSE', 'MAE'], cv=3, verbose=True)

#IMPORTACION EXCEL
from pandas import read_excel,DataFrame
ETFs = read_excel('ETFs(excel).xlsx',index_col=0)
ETFs.reset_index(inplace = True)

#GENERANDO RECOMENDACIONES
import difflib
import random
```

Importación de librerías.

Código 5-22: Filtro colaborativo SVD.


```
def get ETF_info(ETFname):  
    ETF_info =ETFs[ETFs['ETFname'] == ETFname][['ETFname', 'Name', 'Descrip', 'Asset Class', 'Asset Class Size','Strategy ', 'Region ', 'ESG Score Global Percentile (%)',  
        'Revenue Exposure to Environmental Impact (%)', 'Revenue Exposure to Social Impact (%)', 'Severe Governance Controversies (%)']]  
    st.write(ETF_info)  
    return ETF_info.to_dict(orient='records')
```

Función que extrae la información de un solo ETF.

Código 5-23: Función que permite extraer las características de un ETF.

```
def predict_review(username, ETFname):  
    """  
    Predicts the review (on a scale of 1-5) that a user would assign to a specific book.  
    """  
    review_prediction = svd.predict(uid=username, iid=ETFname)  
    return review_prediction.est  
  
def generate_recommendation(username, model, thresh=2):  
    """  
    Generates a ETF recommendation for a user based on a rating threshold. Only  
    ETFs with a predicted rating at or above the threshold will be recommended  
    """  
    ETFs_names = List(ETFs['ETFname'].values)  
    random.shuffle(ETFs_names)  
  
    for ETF in ETFs_names:  
        rating = predict_review(username, ETF, model)  
        #st.write(rating)  
        if rating >= thresh:  
            ETFname = ETF  
            get ETF_info(ETFname)
```

Código 5-24: Función que despliega las recomendaciones

5.4.6 SISTEMA HÍBRIDO 1

Finalmente se construye un sistema híbrido que es la recomendación final que verá el usuario junto con la lista de los 5 ETFs más seleccionados.

Este algoritmo no es más que el uso conjunto del sistema basado en la **descripción y el SVD**. Funciona de la siguiente forma:

- De los 5 ETFs puntuados por el usuario se selecciona aquel o aquellos con la nota más alta.
- A estos se les aplica el sistema basado en la descripción para encontrar aquellos más similares.

- Se crea una lista con estos últimos y se les aplica el SDV para predecir las puntuaciones que dicho usuario les otorgaría.
- Se ofrecen como recomendación aquellos tres para los que se hayan obtenidos mejores puntuaciones.

```
def hybrid_recom(username):
    #obtenemos en formato df las puntuaciones del usuario
    ratings = sqt.check_userspuntuations(username)
    ratings.columns=['ETFname','Puntuacion']
    # ordenamos de mayo a menor las puntuaciones
    raitings = ratings.sort_values('Puntuacion',ascending = False)
    st.write('Segun estas puntuaciones que nos has dado:')
    st.write(raitings)
    # seleccionamos aquellos con mejor puntuacion
    lista = [raitings.iloc[0,0]]
    for n in range(0,4):
        if raitings.iloc[0,1] == raitings.iloc[n+1,1]:
            lista.append(raitings.iloc[n+1,0])
    #Empleamos el content-base recomendar a los ETFs seleccionados
    recomendaciones = []
    for item in lista:
        # aplicamos el sistema de recomendación basado en la descripción
        recom = content_recommender(item)
        recomendaciones.append((recom.loc[:, 'ETFname']).tolist())
    #creamos una lista con los resultados anteriores
    flat_list = list(itertools.chain(*recomendaciones))
    # creamos un Dataframe con los nombre de los ETFs y el número de veces que
    #han aparecido en las recomendaciones anteriores
    (unique, counts) = np.unique(flat_list, return_counts = True)
    tabla = {
        'Nombre': unique,
        'count':counts }
    tabla = pd.DataFrame(tabla).sort_values('count',ascending = False)
    # escogemos los 4 mas repetidos en las recomendaciones
    recom_final = tabla.iloc[0:5,0]
    # aplicamos SVD a los 5 ETFs escogidos anteriormente y los presentamos en orden descendente
    st.write('Creemos que estas opciones te pueden interesar:')
    ETfs_names = recom_final.tolist()
    punt_est = []
    for ETF in recom_final:
        ETF
        rating = sv.predict_review('patricia2', ETF)
        punt_est.append(rating)
        ETFname = ETF
        sv.get ETF_info(ETFname)
    tabla_final = {
        'Nombre': ETfs_names,
        'Punt_est':punt_est }
    tabla_final = pd.DataFrame(tabla_final).sort_values('Punt_est',ascending = False)
    st.write(tabla_final)
```

De las valoraciones que ha hecho el usuario se extraen aquellas con la mejor puntuación y se insertan en una lista.

A los ETFs de esta lista se les aplica el sistema de recomendación basado en la descripción y se escogen las tres primeras recomendaciones para cada uno de ellos, creando una nueva lista con ellos.

Se aplica a la lista anterior el algoritmo de SVD y se ofrece como recomendación al usuario en orden descendente según la puntuación estimada.

Código 5-25: Sistema Híbrido 1

5.4.7 SISTEMA HÍBRIDO 2

El siguiente sistema es el mismo que para el caso anterior, con la diferencia de que ahora se combinan los algoritmos descritos en los puntos 5.6.4 y 5.6.6 .

```
def hybrid(username):
    #obtenemos en formato df las puntuaciones del usuario
    ratings = sqt.check_userspuntuations(username)
    ratings.columns=['ETFname','Puntuacion']
    #ordenamos de mayo a menor las puntuaciones
    raitings = ratings.sort_values('Puntuacion',ascending = False)
    #seleccionamos aquellos con mejor puntuacion
    lista = [raitings.iloc[0,0]]
    for n in range(0,4):
        if raitings.iloc[0,1] == raitings.iloc[n+1,1]:
            lista.append(raitings.iloc[n+1,0])
    #Empleamos el content-base recomendar a los ETFs seleccionados
    recomendaciones = []
    for item in lista:
        #aplicamos el sistema de recomendación basado en la decripción
        recom = content_recommender(item)
        recomendaciones.append((recom.loc[:,'ETFname']).tolist())
    #creamos una lista con los resultados anteriores
    flat_list = list(itertools.chain(*recomendaciones))
    #creamos un Dataframe con los nombre de los ETFs y el número de veces que
    #han aparecido en las recomendaciones anteriores
    (unique, counts) = np.unique(flat_list, return_counts = True)
    tabla = {
        'Nombre': unique,
        'count':counts }
    tabla = pd.DataFrame(tabla).sort_values('count',ascending = False)
    #escogemos los 4 mas repetidos en las recomendaciones
    recom_final = tabla.iloc[0:5,0]
    #aplicamos SVD a los 5 ETFs escogidos anteriormente y los presentamos en orden descendente
    st.write('Segun las puntuaciones que nos has dado Creemos que estas opciones te pueden interesar:')
    ETFs_names = recom_final.tolist()
    punt_est = []
    for ETF in recom_final:
        ETF
        rating = rc.cf_user_wmean(username, ETF)
        punt_est.append(rating)
        ETFname = ETF
        sv.get ETF_info(ETFname)
    tabla_final = {
        'Nombre': ETFs_names,
        'Punt_est':punt_est }
    tabla_final = pd.DataFrame(tabla_final).sort_values('Punt_est',ascending = False)
    st.write(tabla_final)
```

Código 5-26: Sistema Híbrido 2.

5.5 CÓDIGO COLOCACIÓN DE ACTIVOS

Para la colocación de activos se emplean varios scripts diferentes que se importaran desde el script principal para su uso. Estos son:

- **Script1 : Efficient Frontier**
- **Script 2: MVP y HRP usando la librería *pypfopt***
- **Script 3: Backtest MVP vs HRP**

A continuación, se presentará el código y una explicación detallada de este para los tres casos:

➤ Script 1 Efficient Frontier:

Dentro de este script se encuentran dos funciones principales:

1. **RandomAllo():**

```

def RandomAllo(selected ETFs):
    st.write('EFFICIENT FRONTIER ALLOCATION')
    #selected ETFs -> nos devuelve una lista de listas
    flat_list = itertools.chain(*selected ETFs)
    selected ETFs = list(flat_list)
    Select_returns = pd.DataFrame()
    retornos = read_excel('Retornos.xlsx', index_col=0)

    #filtramos en la tabla aquellos ETFs que han sido seleccionados por el usuario
    for ETF in retornos:
        for selc in selected ETFs :
            if selc == ETF:
                Select_returns =Select_returns.append(retornos[ETF])

    Select_returns=Select_returns.transpose()
    st.dataframe(Select_returns)
    # calculamos la matriz de correlación de las diferentes empresas
    #para ver en que medida estan correlacionadas
    corr = Select_returns.pct_change(1).corr()
    st.write('Correlation matrix')
    st.dataframe(corr)

```

Transformamos en una lista los ETFs seleccionados por el usuario y descargamos los retomos históricos.

Obtenemos matriz de correlaciones.

Código 5-27: Función iterative que crea diferentes portfolios asignados pesos aleatorios.

```
# Fill the missing values with the last value available in the dataset.
Select_returns=Select_returns.fillna(method='ffill')
#Retornos normalizados de los últimos 5 años
stoks_fig = plt.figure()
stock_normed = (Select_returns/Select_returns.iloc[0]).plot()
# st.pyplot(stoks_fig)
mean_daily_ret = Select_returns.pct_change(1).mean()*252
log_ret = np.log(Select_returns/Select_returns.shift(1))
np.random.seed(100)
num_ports = 10000

all_weights = np.zeros((num_ports,len(Select_returns.columns)))
ret_arr = np.zeros(num_ports)
vol_arr = np.zeros(num_ports)
sharpe_arr = np.zeros(num_ports)

for ind in range(num_ports):
    # Create Random Weights
    weights = np.array(np.random.random(5))
    # Rebalance Weights
    weights = weights / np.sum(weights)
    # Save Weights
    all_weights[ind,:] = weights
    # Expected Return
    ret_arr[ind] = np.sum((log_ret.mean() * weights) *252)
    # Expected Variance
    vol_arr[ind] = np.sqrt(np.dot(weights.T, np.dot(log_ret.cov() * 252, weights)))
    # Sharpe Ratio
    sharpe_arr[ind] = ret_arr[ind]/vol_arr[ind]
```

Calculamos los
retornos diarios.

**Creamos 10000 posibles
colocaciones de activos
diferentes de forma
aleatoria.**

Empleando Retornos
logarítmicos en este caso.

Código 5-28

```
opt = sharpe_arr.argmax()
final_weights_ite = all_weights[opt,:]
st.write('Pesos finale proceso iterativo')
st.table(final_weights_ite)
st.write('SR')
st.write(sharpe_arr[opt])
max_sr_ret = ret_arr[opt]
max_sr_vol = vol_arr[opt]
#Plot Eff
st.write('scatter plot')
scater = plt.figure(figsize=(12,8))
plt.scatter(vol_arr,ret_arr,c=sharpe_arr,cmap='plasma')
plt.colorbar(Label='Sharpe Ratio')
plt.xlabel('Volatility')
plt.ylabel('Return')
plt.scatter(max_sr_vol,max_sr_ret,c='red',s=50,edgecolors='black')
st.pyplot(scater)
```

Obtenemos los valores del
portafolio más optimo en
cuanto a sharpe ratio.

Representamos todos estos
portafolios empleando un
diagrama de dispersión.

Código 5-29: Extracción del portafolio óptimo dentro de los anteriores.

Ejemplo:

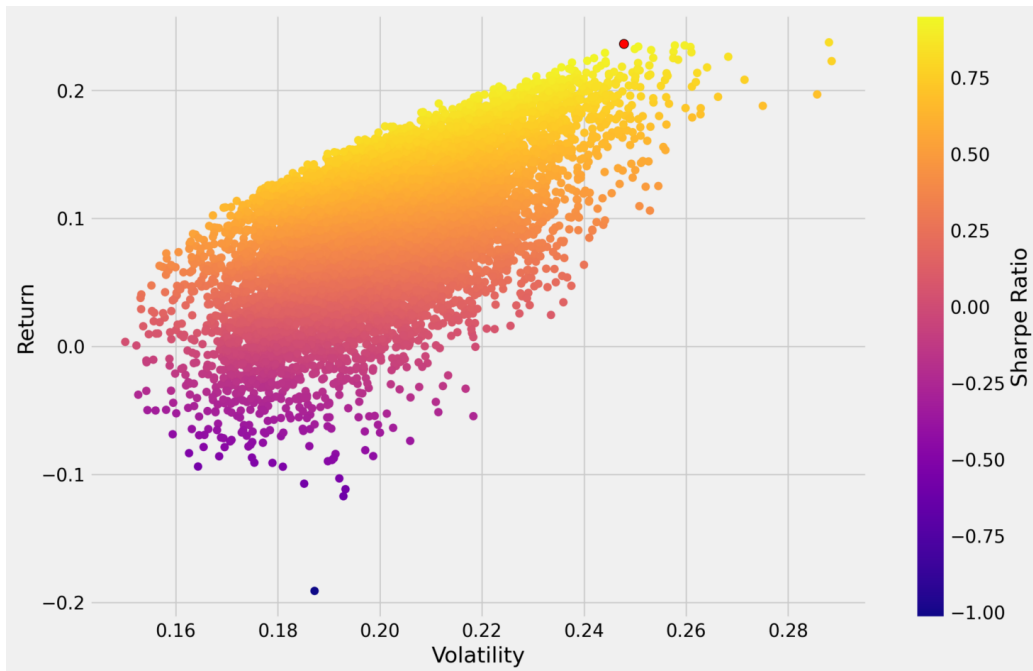


Figure 5-3: Ejemplo del gráfico de dispersión de un usuario para 1000 portafolios diferentes

2. Opt_eff():

Esta función lleva a cabo un problema de optimización, donde se busca minimizar el negativo del sharpe ratio, es decir maximizar el mismo. Para encontrar así la colocación de activos óptima según MVP.

```
def opt_eff(selected ETFs):
    #selected ETFs -> nos devuelve una lista de listas
    flat_list = itertools.chain(*selected ETFs)
    selected ETFs = list(flat_list)
    Select_returns = pd.DataFrame()
    retornos = read_excel('Retornos.xlsx', index_col=0)
    columns_names = retornos.columns.values
    columns_names_list = list(columns_names)
    #filtramos en la tabla aquellos ETFs que han sido seleccionados por el usuario
    for ETF in retornos:
        for selc in selected ETFs :
            if selc == ETF:
                Select_returns =Select_returns.append(retornos[ETF])

    Select_returns=Select_returns.transpose()
    corr = Select_returns.pct_change(1).corr()
    # Fill the missing values with the last value available in the dataset.
    Select_returns=Select_returns.fillna(method='ffill')
    #Retornos normalizados de los últimos 5 años
    stoks_fig = plt.figure()
    stock_normed = (Select_returns/Select_returns.iloc[0])
    mean_daily_ret = Select_returns.pct_change(1).mean()*252
    log_ret = np.log(Select_returns/Select_returns.shift(1))

def get_ret_vol_sr(weights):
    """
    Takes in weights, returns array or return,volatility, sharpe ratio
    """
    weights = np.array(weights)
    ret = np.sum(log_ret.mean() * weights) * 252
    vol = np.sqrt(np.dot(weights.T, np.dot(log_ret.cov() * 252, weights)))
    sr = ret/vol
    return np.array([ret,vol,sr])

def neg_sharpe(weights):
    return get_ret_vol_sr(weights)[2] * -1

def check_sum(weights):
    """
    Returns 0 if sum of weights is 1.0
    """
    return np.sum(weights) - 1
```

Obtenemos retornos históricos de ETFs seleccionados por el usuario.

Hacemos la matriz de correlaciones entre ellos.

Calculamos los retornos diarios, los normalizamos y obtenemos los retornos logarítmicos.

Función que devuelve el retorno, la volatilidad y el sharpe ratio de un portafolio a partir de sus pesos.

Función que devuelve el valor negativo del shape ratio.

Función que sirve para comprobar que los pesos de los elementos que componen la cartera suman uno.

Código 5-30.

```

cons = ({'type':'eq', 'fun': check_sum})
# 0-1 bounds for each weight
bounds = ((0, 1), (0, 1), (0, 1), (0, 1),(0, 1))
# Initial Guess (equal distribution)
init_guess = [0.2,0.2,0.2,0.2,0.2]
# Sequential Least Squares Programming (SLSQP).
opt_results = minimize(neg_sharpe,init_guess,method='SLSQP',bounds=bounds,constraints=cons)

columns_names = Select_returns.columns.values
columns_names_list = list(columns_names)
nombres = columns_names_list

final_weights_opt =opt_results.x
st.write('Los pesos óptimos para su cartera según MVP')
final_weights = pd.DataFrame(final_weights_opt,index =nombres ,columns =['Pesos'])
st.table(final_weights)
vec_result= get_ret_vol_sr(opt_results.x)
vec_result = pd.DataFrame(vec_result,index = ['Return','Volatility','Sharpe Ratio'],columns = ['Valores'])
st.table(vec_result)

st.write('Resultados')
pie_chart = plt.figure(figsize=(12,8))
plt.pie(final_weights_opt,Labels = nombres)
st.pyplot(pie_chart)

stoks_fig = plt.figure(figsize=(12,8))
stock_normed = (Select_returns/Select_returns.iloc[0])
plt.plot(stock_normed)
st.line_chart(stock_normed)

```

Se lleva a cabo el problema de optimización utilizando el método 'SLSQP'(sequential least squares programming algorithm)

Se escriben y presentan los resultados y gráficos.

Código 5-31: Obtención del portafolio óptimo mediante un problema de optimización.

```

ETF1 =pd.DataFrame(Select_returns.iloc[:,0])
ETF2 =pd.DataFrame(Select_returns.iloc[:,1])
ETF3 =pd.DataFrame(Select_returns.iloc[:,2])
ETF4 =pd.DataFrame(Select_returns.iloc[:,3])
ETF5 =pd.DataFrame(Select_returns.iloc[:,4])
ETF1.columns = ['Adj Close']
ETF2.columns = ['Adj Close']
ETF3.columns = ['Adj Close']
ETF4.columns = ['Adj Close']
ETF5.columns = ['Adj Close']
columns_names = ETF1.columns.values

st.subheader('Evolución histórica de la cartera (10000)')

for stock_df in (ETF1,ETF2,ETF3,ETF4,ETF5):
    stock_df['Normed Return'] = stock_df['Adj Close']/stock_df.iloc[0]['Adj Close']

for stock_df,allo in zip([ETF1,ETF2,ETF3,ETF4,ETF5],final_weights_opt.tolist()):
    stock_df['Allocation'] = stock_df['Normed Return']*allo

#evaluemos suponiendo que introducimos 10000 dolares
for stock_df in (ETF1,ETF2,ETF3,ETF4,ETF5):
    stock_df['Position Values'] = stock_df['Allocation']*10000

portfolio_val = pd.concat([ETF1['Position Values'],ETF2['Position Values'],ETF3['Position Values'],ETF4['Position Values'],ETF5['Position Values']],axis=1)
portfolio_val.columns = ['ETF1 Pos','ETF2 Pos','ETF3 Pos','ETF4 Pos','ETF5 Pos']

port_return = plt.figure(figsize=(20,10))
portfolio_val['Total Pos'] = portfolio_val.sum(axis=1)
plt.plot(portfolio_val['Total Pos'])
plt.title('Total Portfolio Value')
st.line_chart(portfolio_val['Total Pos'])
st.write(portfolio_val['Total Pos'])

```

Se calcula y se presenta la evolución histórica de dicho portafolio, para una cartera de 10000 euros.

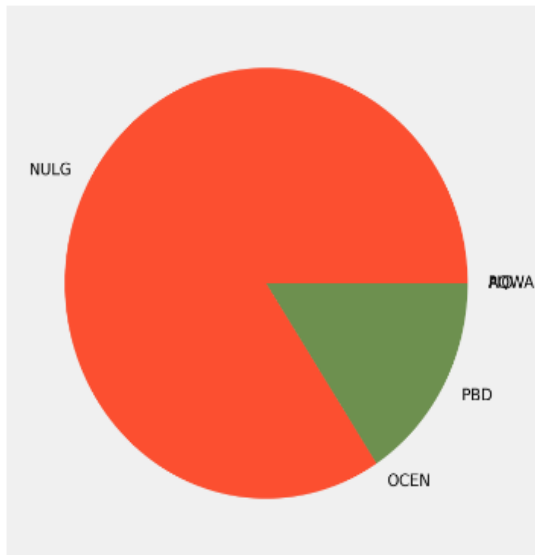
Código 5-32: Evolución histórica de una cartera con los pesos obtenidos para el mejor portafolio.

Ejemplo:

	Pesos
AQWA	0.0000
WULLO	0.2428
OCEN	0.0000
PBD	0.1577
PJO	0.0000

	Valores
Return	0.2490
Volatility	0.2440
Sharpe Ratio	1.0040

Resultados



Evolución histórica de la cartera (10000)



	Total Pos
2019-01-02T00:00:00	10,000.0000
2019-01-08T00:00:00	9,788.0017
2019-01-14T00:00:00	10,127.8489
2019-01-20T00:00:00	10,284.5410
2019-01-26T00:00:00	10,309.8177
2019-01-31T00:00:00	10,410.5888
2019-02-05T00:00:00	10,480.4088
2019-02-11T00:00:00	10,441.6399
2019-02-17T00:00:00	10,834.7270
2019-01-15T00:00:00	10,498.9440

Figure 5-4: Ejemplo resultado de optimización MVP para la cartera de un usuario.

➤ **Script 2 MVP y HRP usando la librería pypfopt:**

En este script se realizan los cálculos para obtener el portafolio más óptimo empleando los métodos MVP y HRP, pero en esta ocasión empleando una librería de Python prediseñada para ello.


```
def HRP_Eff(selected ETFs):

    flat_list = chain(*selected ETFs)
    selected ETFs = List(flat_list)
    Select_returns = pd.DataFrame()
    retornos = read_excel('Retornos.xlsx', index_col=0)
    columns_names = retornos.columns.values
    columns_names_list = List(columns_names)
    #filtramos en la tabla aquellos ETFs que han sido seleccionados por el usuario
    for ETF in retornos:
        for selc in selected ETFs :
            if selc == ETF:
                Select_returns =Select_returns.append(retornos[ETF])

    Select_returns=Select_returns.transpose()
    #st.dataframe(Select_returns)
    # Fill the missing values with the last value available in the dataset.
    Select_returns=Select_returns.fillna(method='ffill')
    #calculate expected annualized return and covariance metrics
    mu = expected_returns.mean_historical_return(Select_returns)
    S = risk_models.sample_cov(Select_returns)
    #optimize maximal sharpe ratio
    ef = efficient_frontier.EfficientFrontier(mu,s)#creal el objeto efficient frontier
    weights = ef.max_sharpe()
    cleaned_weights = ef.clean_weights()
    pesos_MVP = pd.DataFrame(List(cleaned_weights.items()),columns = ['ETFs cartera', 'pesos'])
    pesos_MVP.set_index(['ETFs cartera'])
    st.write('Efficiente weights')
    st.table(pesos_MVP)

    ef.portfolio_performance(verbose = True )
    # Pie chart Eff
    st.write('Pie char Eff')
    pie_chart = plt.figure(figsize=(12,8))
    nombres = List(cleaned_weights.keys())
    weights_eff = List(cleaned_weights.values())
    plt.pie(weights_eff, Labels = nombres)
    st.pyplot(pie_chart)
```

Cálculo del portfolio más eficiente. empleando MVP.

Presentación de gráficos y resultados.

Código 5-33: Cálculo del portafolio óptimo con MVP empleando librería 'pypfopt'

```
# Hierarchical risk parity
hrp = HRPOpt(returns)
hrp.optimize()
weights = hrp.clean_weights()
st.write('HRP weights')
pesos = pd.DataFrame(List(weights.items()),columns = ['ETFs cartera', 'pesos'])
pesos.set_index(['ETFs cartera'])
st.table(pesos)
#Pie Chart
#Plot portfolios

# Pie chart HRP
st.write('Pie char HRP')
pie_chart = plt.figure(figsize=(12,8))
nombres = List(weights.keys())
weights_hrp = List(weights.values())
plt.pie(weights_hrp, Labels = nombres)
st.pyplot(pie_chart)

w_2 = pd.Series(weights).to_frame().sort_values(by=0, ascending=False, axis=0)
hrp.portfolio_performance(verbose=True)
figure = plt.figure(figsize=(20,8))
plotting.plot_dendrogram(hrp) # to plot dendrogram
st.pyplot(figure)
```

Cálculo del portfolio más eficiente empleando HRP.

Presentación de gráficos y resultados.

Código 5-34: Cálculo del portafolio óptimo con HRP empleando librería 'pypfopt'

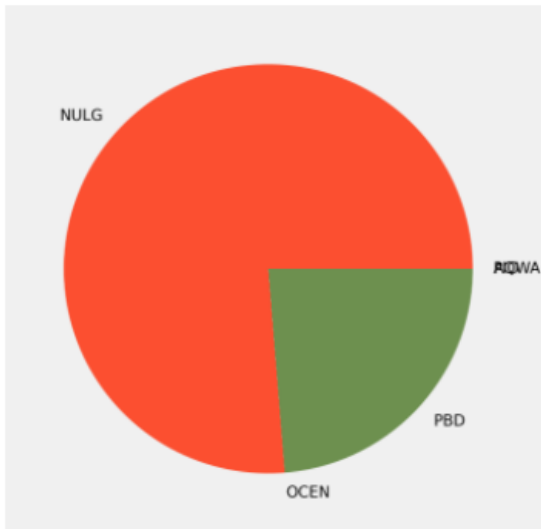
Ejemplo:

	ETFs cartera	pesos
0	AQWA	0.0000
1	NULG	0.7629
2	OCEN	0.0000
3	PBD	0.2371
4	PIO	0.0000

HRP weights

	ETFs cartera	pesos
0	AQWA	0.4086
1	NULG	0.1785
2	OCEN	0.1805
3	PBD	0.0879
4	PIO	0.1695

Pie chart EF



Pie chart HRP

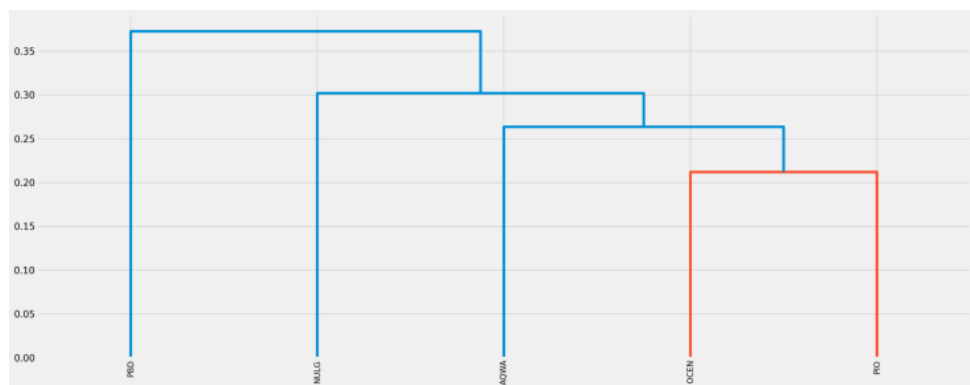
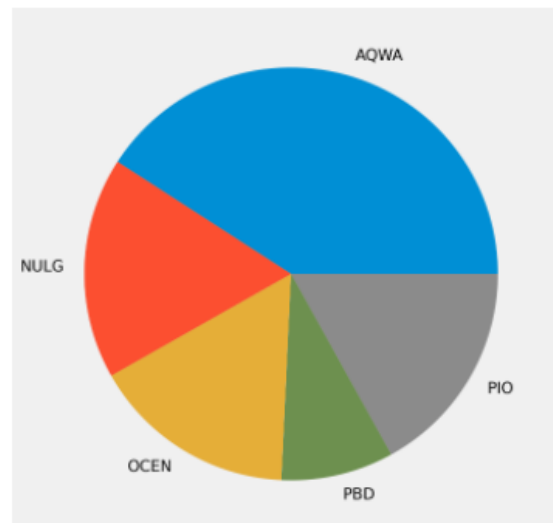


Figure 5-5: Ejemplo optimización cartera usuario.

➤ **Script 3 Backtest MVP vs HRP:**

Este script se empleará para realizar una comparativa de ambos métodos realizando un *backtest* de ambas metodologías en el pasado.

Nota: El código empleado para esta sección ha sido mayoritariamente extraído del libro “Machine Learning and Data Science Blueprints for Finance”

```
def correlDist(corr):
    # A distance matrix based on correlation, where 0<=d[i,j]<=1
    # This is a proper distance metric
    dist = ((1 - corr) / 2.)**.5 # distance matrix
    return dist
#QUASIDIAGONALIZACION
def getQuasiDiag(link):
    # Sort clustered items by distance
    link = link.astype(int)
    sortIx = pd.Series([link[-1, 0], link[-1, 1]])
    numItems = link[-1, 3] # number of original items
    while sortIx.max() >= numItems:
        sortIx.index = range(0, sortIx.shape[0] * 2, 2) # make space
        df0 = sortIx[sortIx >= numItems] # find clusters
        i = df0.index
        j = df0.values - numItems
        sortIx[i] = link[j, 0] # item 1
        df0 = pd.Series(link[j, 1], index=i + 1)
        sortIx = sortIx.append(df0) # item 2
        sortIx = sortIx.sort_index() # re-sort
        sortIx.index = range(sortIx.shape[0]) # re-index
    return sortIx.tolist()
#RECURSIVE BISECTION
def getClusterVar(cov, cItems):
    # Compute variance per cluster
    cov_ = cov.loc[cItems, cItems] # matrix slice
    w_ = getIVP(cov_).reshape(-1,1)
    cVar = np.dot(np.dot(w_.T, cov_), w_)[0,0]
    return cVar
```

Función que convierte las correlaciones en distancias.

Función que hace la Quasi-diagonalization.

Función que calcula la varianza de cada cluster.

Código 5-35

```
def getRecBipart(cov, sortIx):
    # Compute HRP alloc
    w = pd.Series(1, index=sortIx)
    cItems = [sortIx] # initialize all items in one cluster
    while len(cItems) > 0:
        cItems = [i[j:k] for i in cItems for j, k in ((0, len(i) // 2), (len(i) // 2, len(i))) if len(i) > 1]
        for i in range(0, len(cItems), 2): # parse in pairs
            cItems0 = cItems[i] # cluster 1
            cItems1 = cItems[i + 1] # cluster 2
            cVar0 = getClusterVar(cov, cItems0)
            cVar1 = getClusterVar(cov, cItems1)
            alpha = 1 - cVar0 / (cVar0 + cVar1)
            w[cItems0] *= alpha # weight 1
            w[cItems1] *= 1 - alpha # weight 2
    return w
```

Función que calcula la colocación final mediante 'Recursive bisection' basada en la covarianza de los clústeres.

Código 5-36

```
def getIVP(cov, **kargs):
    # Compute the inverse-variance portfolio
    ivp = 1. / np.diag(cov)
    ivp /= ivp.sum()
    return ivp

def getMVP(cov):

    cov = cov.T.values
    n = len(cov)
    N = 100
    mus = [10 ** (5.0 * t / N - 1.0) for t in range(N)]

    # Convert to cvxopt matrices
    S = opt.matrix(cov)
    #pbar = opt.matrix(np.mean(returns, axis=1))
    pbar = opt.matrix(np.ones(cov.shape[0]))

    # Create constraint matrices
    G = -opt.matrix(np.eye(n)) # negative n x n identity matrix
    h = opt.matrix(0.0, (n, 1))
    A = opt.matrix(1.0, (1, n))
    b = opt.matrix(1.0)

    # Calculate efficient frontier weights using quadratic programming
    solvers.options['show_progress'] = False
    portfolios = [solvers.qp(mu * S, -pbar, G, h, A, b)['x']
                  for mu in mus]
    ## CALCULATE RISKS AND RETURNS FOR FRONTIER
    returns = [blas.dot(pbar, x) for x in portfolios]
    risks = [np.sqrt(blas.dot(x, S * x)) for x in portfolios]
    ## CALCULATE THE 2ND DEGREE POLYNOMIAL OF THE FRONTIER CURVE
    m1 = np.polyfit(returns, risks, 2)
    x1 = np.sqrt(m1[2] / m1[0])
    # CALCULATE THE OPTIMAL PORTFOLIO
    wt = solvers.qp(opt.matrix(x1 * S), -pbar, G, h, A, b)['x']

    return list(wt)
```

Función que calcula “The inverse-variance portfolio”

Función que realiza la optimización de la cartera mediante MVP usando clustering, a partir de la matriz de covarianzas.

Código 5-37

```
def getHRP(cov, corr):
    # Construct a hierarchical portfolio
    dist = correlDist(corr)
    link = sch.linkage(dist, 'single')
    sortIx = getQuasiDiag(link)
    sortIx = corr.index[sortIx].tolist()
    hrp = getRecBipart(cov, sortIx)
    return hrp.sort_index()

def get_all_portfolios(returns):
    cov, corr = returns.cov(), returns.corr()
    hrp = getHRP(cov, corr)
    mvp = getMVP(cov)
    mvp = pd.Series(mvp, index=cov.index)
    return portfolios
```

Función que convina los pasos de: clustering, quasi-diagonalization y recursive bisection para obtener los pesos finales de la cartera según HRP.

Función que devuelve los resultados de los portfolios

Código 5-38

```
def EjecucionHRP(selected ETFs):
    #Diable the warnings
    flat_list = itertools.chain(*selected ETFs)
    selected ETFs = list(flat_list)
    Select_returns = pd.DataFrame()
    retornos = read_excel('Retornos.xlsx', index_col=0)
    import warnings
    columns_names = retornos.columns.values
    columns_names_list = list(columns_names)
    #filtramos en la tabla aquellos ETFs que han sido seleccionados por el usuario
    for ETF in retornos:
        for selc in selected ETFs :
            if selc == ETF:
                Select_returns =Select_returns.append(retornos[ETF])

    warnings.filterwarnings('ignore')
    # peek at data
    set_option('display.width', 100)
    #Checking for any null values and removing the null values'''
    print('Null Values =',Select_returns.isnull().values.any())
    #eliminamos las columnas con mas de un 30% de valores nulos
    missing_fractions = Select_returns.isnull().mean().sort_values(ascending=False)
    Select_returns = Select_returns.transpose()

    drop_list = sorted(list(missing_fractions[missing_fractions > 0.3].index))

    Select_returns.drop(labels=drop_list, axis=1, inplace=True)
    st.write(Select_returns)

    # Fill the missing values with the last value available in the dataset.
    Selected ETFs=Select_returns.fillna(method='ffill')
    Selected ETFs.head(2)

    X= Selected ETFs.copy('deep')
    row= len(X)
    train_len = int(row*.8)

    X_train = X.head(train_len)
    X_test = X.tail(row-train_len)

    #Calculate percentage return
    returns = X_train.pct_change().dropna()
    returns_test=X_test.pct_change().dropna()
    #Calculate linkage
    dist = correlDist(returns.corr())
    link = linkage(dist, 'ward') #
    link[0]
    #Plot portfolios
    portfolios = get_all_portfolios(returns)
    st.title('Portfolios')
    st.table(portfolios)
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(30,20))
    ax1.pie(portfolios.iloc[:,0], labels = portfolios.index )
    ax1.set_title('MVP', fontsize = 30)
    ax2.pie(portfolios.iloc[:,1], labels = portfolios.index )
    ax2.set_title('HRP', fontsize = 30)
    st.pyplot(fig)
    #BACKTESTING

    Insample_Result=pd.DataFrame(np.dot(returns,np.array(portfolios)), \
                                columns=['MVP', 'HRP'], index = returns.index)
    OutOfSample_Result=pd.DataFrame(np.dot(returns_test,np.array(portfolios)), \
                                   columns=['MVP', 'HRP'], index = returns_test.index)
    st.line_chart(Insample_Result.cumsum())
    st.line_chart(OutOfSample_Result.cumsum())

    #In_sample Results
    st.write('In Sample Results')
    stddev = Insample_Result.std() * np.sqrt(252)
    sharp_ratio = (Insample_Result.mean()*np.sqrt(252))/(Insample_Result).std()
    Results = pd.DataFrame(dict(stddev=stddev, sharp_ratio = sharp_ratio))
    st.table(Results)

    #Outof_sample Results
    st.write('Out of Sample results')
    stddev_oos = OutOfSample_Result.std() * np.sqrt(252)
    sharp_ratio_oos = (OutOfSample_Result.mean()*np.sqrt(252))/(OutOfSample_Result).std()
    Results_oos = pd.DataFrame(dict(stddev_oos=stddev_oos, sharp_ratio_oos = sharp_ratio_oos))
    st.table(Results_oos)
```

Descarga de retornos de los ETFs seleccionados por el usuario.

Dividimos los datos de los retornos en *train* and *test* set.(80-20%)

Dividimos los datos de los retornos en *train* and *test* set.(80-20%)

Presentación de los portfolios en gráficos.

Back test del rendimiento de los portfolios tanto para los datos usados para el entrenamiento del modelo como los usados para el test.

Ejemplo:

Portfolios

	MVP	HRP
AQWA	0.4210	0.1901
NULG	0.3153	0.2306
PIO	0.1970	0.1996
VEGI	0.0634	0.2056
XLK	0.0033	0.1741

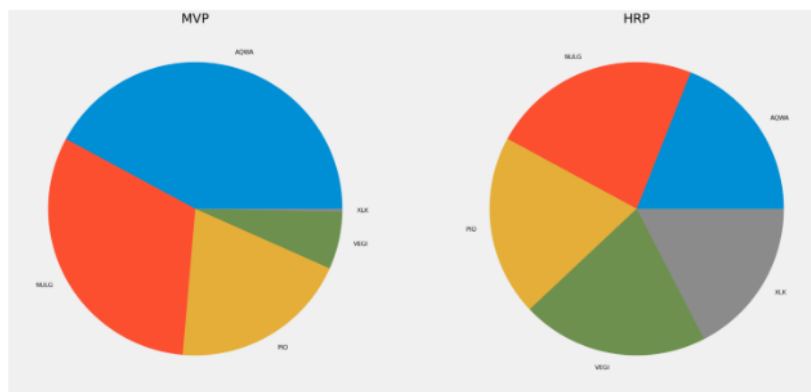


Figure 5-6: Ejemplo portafolios MVP y HRP usuario.



Figure 5-7: Ejemplo Backtest MVP vs HRP

In Sample Results

	stdev	sharp_ratio
MVP	0.1126	2.5634
HRP	0.1177	1.9954

Out of Sample results

	stdev_oos	sharp_ratio_oos
MVP	0.1628	-0.6322
HRP	0.1656	0.0036

Figure 5-8: Comparación resultados del desempeño “in sample” vs “out of sample”

5.6 CÓDIGO CÁLCULO JUBILACIÓN

El código de este apartado se encuentra en el archivo llamado **Jubilacion.py** y que se importa desde el script principal para su uso.

Dentro de este tenemos dos funciones que se utilizarán para hacer los dos cálculos que se ofrecen como opción a los usuarios.

1. Cálculo 1:

En esta primera opción el usuario introduce una serie de variables para simular un escenario y se grafica la evolución de su patrimonio según dicho escenario.

```
def jubilacion():
    st.title('Retirement Simulation')
    Edad = st.number_input(' Introduce edad actual ',value =20)
    Edad_inicial = Edad
    Edad_final = st.number_input('Introduce la edad hasta la que vivirás ',value =80)
    Edad_Jubilacion = st.number_input('Introduce edad de jubilación',value =50)
    invested_years = Edad_Jubilacion - Edad
    Años_necesarios =Edad_final-Edad_Jubilacion #cantidad de años que necesito vivir de las inversiones
    Años_totales = Edad_final-Edad
    Inflacon = st.number_input('Introduce % de inflación')
    interest_rate = st.number_input('Introduce % de rentabilidad anual ')
    Gasto_anual = st.number_input('Introduce cantidad de dinero que necesitas anualmente para vivir ', value =20000)
    deposito_inicial = st.number_input('Introduce deposito inicial ')
    aportaciones_anuales = st.number_input('Introduce aportación anual ')
    money_evolution = deposito_inicial

    Year = []
    Money_evolution = []

    Year.append(Edad)
    Money_evolution.append(money_evolution)
    j=0
    for i in range(0,Años_totales):
        if ((Edad) < Edad_Jubilacion):
            money_evolution =(money_evolution + aportaciones_anuales*j)*(1-Inflacon/100)*(1+interest_rate/100)
            Edad = Edad + 1
            Year.append(Edad)
            Money_evolution.append(money_evolution)
            j=j+1
        else:
            money_evolution = (money_evolution - Gasto_anual*(1+Inflacon/100))*(1+(interest_rate-2)/100)# consideramos
            Edad = Edad + 1
            Year.append(Edad)
            Money_evolution.append(money_evolution)
```

Se piden datos al usuario.

Cálculo de la evolución del patrimonio.

Código 5-41: Función para el cálculo de la evolución del dinero.

Ejemplo:

Intrduce edad actual

24	-	+
----	---	---

Intrduce la edad hasta la que vivirás

90	-	+
----	---	---

Intrduce edad de jubilación

65	-	+
----	---	---

Intrduce % de inflación

3,00	-	+
------	---	---

Intrduce % de rentabilidad anual

4,50	-	+
------	---	---

Intrduce cantidad de dinero que necesitas anualmente para vivir

30000	-	+
-------	---	---

Intrduce deposito inicial

10000,00	-	+
----------	---	---

Intrduce aportación anual

10000,00	-	+
----------	---	---

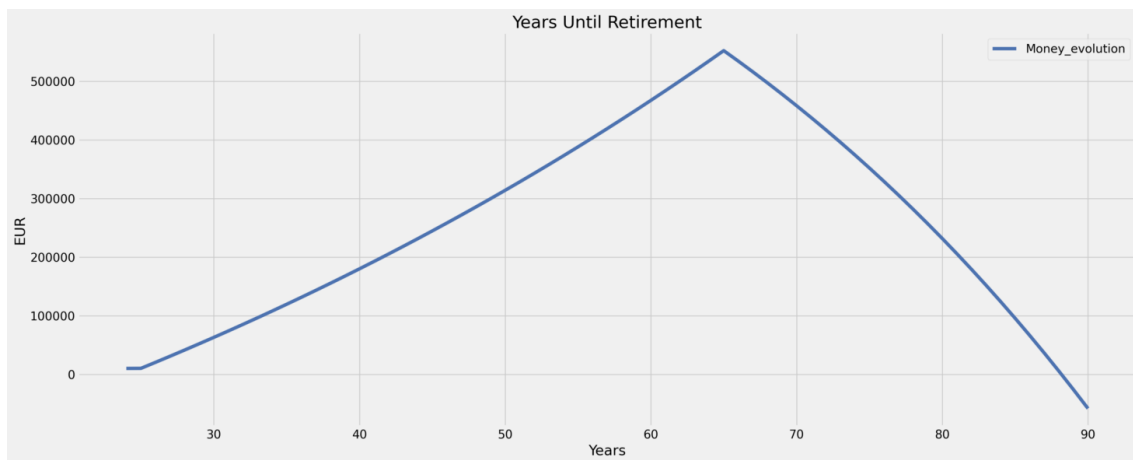


Figure 5-9: Ejemplo Cálculo 1 (Retirement Simulation)

NOTA: Se dividen los años de vida restantes en dos. Los años previos a la jubilación y los posteriores. Para esto últimos se ha considerado que la rentabilidad anual esperada de las inversiones es un 2% inferior a la introducida, ya que en esta etapa se suele optar por una inversión más conservadora.

```
df = pd.DataFrame()
df['Money_evolution'] = Money_evolution
df['Edad'] = Year
df.set_index('Edad', inplace = True)
st.dataframe(df)

figure = plt.figure(figsize=(20,8))
plt.plot(df['Money_evolution'], Label = 'Money_evolution')
plt.title('Years Until Retirement')
plt.xlabel('Years')
plt.ylabel('EUR')
#plt.xticks(df['Edad'], rotation= 45)
plt.legend()
st.pyplot(figure)
```

Presentación de resultados : tabla +gráfica.

Código 5-42: Presentación gráfica de resultados.

2. Cálculo 2:

En esta opción lo que se ofrece como resultado es la cuota anual de inversión necesaria para lograr el objetivo del usuario según el escenario establecido por el mismo.

```
def jubilacion2():
    st.title('Retirement Simulation')
    Edad = st.number_input('Introduce edad actual',value = 20)
    Edad_inicial = Edad
    Edad_final = st.number_input('Introduce edad hasta la que vivirás',value =80)
    Edad_Jubilacion = st.number_input('Introduce edad de jubilación',value =60)
    invested_years = Edad_Jubilacion - Edad
    Años_necesarios =Edad_final -Edad_Jubilacion #cantidad de años que necesito vivir de las inversiones
    Años_totales = Edad_final-Edad
    int(Años_totales)
    Inflacion = st.number_input(' Introduce % de inflación')
    interest_rate = st.number_input(' Introduce % de rentabilidad anual ')
    Gasto_anual = st.number_input(' Introduce cantidad de dinero que necesitas anualmente para vivir (en dinero actual) ',value = 20000)
    deposito_inicial = st.number_input('Introduce deposito inicial ')
    money_evolution = deposito_inicial
```

Se piden datos al usuario.

Código 5-43

```
Year = []
sueldo = []
factor = []
ahorro_jubilacion = []
contador = abs(Edad-Edad_Jubilacion)

for i in range(0,Años_totales):
    contador = abs(Edad-Edad_Jubilacion)
    if Edad == Edad_inicial:
        sueldo.append(Gasto_anual)
    else:
        sueldo1 = sueldo[i-1]*(1+Inflacion/100)
        sueldo.append(sueldo1)

    if ((Edad) < Edad_Jubilacion):
        Year.append(Edad)
        factor1 = (1+interest_rate/100)**contador
        contador = abs(Edad-Edad_Jubilacion)
        factor.append(factor1)

    else:
        Year.append(Edad)
        ahorro_jubilacion1 = sueldo1/((1+interest_rate/100)**contador)
        ahorro_jubilacion.append(ahorro_jubilacion1)

    Edad = Edad + 1

Cantidad_necesaria = sum(ahorro_jubilacion)
suma_factor = sum(factor)
st.subheader('La cantidad que enecesitamos tener a la edad de jubilación es:')
st.metric(Label = '',value = round(Cantidad_necesaria,2))
cuota = Cantidad_necesaria/suma_factor
st.subheader('La cuota anual que necesitamos invertir para conseguir la cifra deseada es :')
st.metric(Label = '',value =round(cuota,2))
```

Cálculos.

Presentación resultados.

Código 5-44: Cálculo jubilación 2.

Ejemplo:

Intruduce edad actual

20 - +

Intruduce edad hasta la que vivirés

80 - +

Intruduce edad de jubilación

65 - +

Intruduce % de inflación

3,00 - +

Intruduce % de rentabilidad anual

4,50 - +

Intruduce cantidad de dinero que necesitas anualmente para vivir (en dinero actual)

30000 - +

Intruduce deposito inicial

10000,00 - +

La cantidad que enecesitamos tener a la edad de jubilación es:

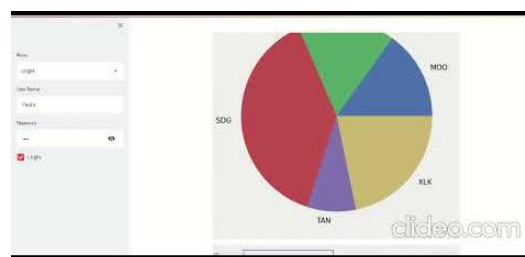
1540923.73

La cuota anual que necesitamos invertir para conseguir la cifra deseada es :

10619.87

Figure 5-10: Ejemplo Cálculo 2 (Retiremt Simulation)

Enlace a video ilustrativo del prototipo final de la aplicación:



Capítulo 6. OPORTUNIDADES EMPRESARIALES AL RESPECTO.

En este capítulo se pretende ofrecer una visión de cómo sería el proceso del desarrollo real de una aplicación de este estilo, así como un análisis de la inversión necesaria y rentabilidad que podría esperarse. Con el objetivo de ilustrar como sería la transición del mundo académico al de la innovación empresarial.

6.1 PASOS PARA EL DESARROLLO

El desarrollo de una aplicación Fintech se compone de las siguientes etapas:

1. Fase de descubrimiento (Discovery phase):

Incluye los siguientes aspectos:

- Análisis del mercado.
- Análisis de los competidores
- Definición del cliente objetivo
- Análisis de costes
- Definición de objetivos

2. Fase de diseño:

Durante esta fase se lleva a cabo el diseño y definición de la interfaz de usuario y la experiencia de este y la creación de un prototipo.

3. Fases de desarrollo y testeo (Development and testing phase):

Fase en la que se transforma el diseño en una aplicación real. Se completa la aplicación y se testea en el mercado. Dentro de esta fase los diferentes profesionales que intervienen son:

- Project manager
- Business analyst
- CTO
- Backend developer
- Frontend developer
- UI/UX designer
- Quality assurance tester

4. Lanzamiento de la aplicación y mantenimiento. (Launch and maintenance phase)

Se lanza la aplicación y se comienza a analizar el *feedback* recibido de los usuarios, con el objetivo de hacer los ajustes y mejoras necesarias.

Para asegurar el correcto funcionamiento será necesario llevar a cabo un mantenimiento continuo de la aplicación. Donde se continúa recibiendo el *feedback* por parte de los usuarios, realizando los ajustes y mejoras necesarias y ofreciendo soporte técnico.

5. Mejora e innovación constante

El mundo Fintech es muy competitivo y evoluciona rápidamente, por eso será necesario seguir invirtiendo e innovación y desarrollo buscando la mejora continua para evitar quedarse atrás respecto a los competidores.

6.2 PROYECCIÓN FINANCIERA

A continuación, se presentan algunas estimaciones en cuanto a los costes en los que sería necesario incurrir, así como predicciones de ingresos que se podría esperar obtener.

Se estima que todo el proceso que engloba desde la ideación hasta el desarrollo de la aplicación duraría entre 9 y 12 meses.

Las predicciones que se presentan a continuación están hechas para un periodo de 5 años, periodo a partir del cual se puede esperar una progresión más constante y estable.

6.2.1 PROYECCIÓN COSTES

En este apartado se realizará una predicción aproximada de todos los costes en los que sería necesario incurrir para el desarrollo de una plataforma como la elaborada durante el proyecto.

Dividimos los costes en dos secciones: Costes iniciales y costes operacionales.

- Costes iniciales:

En estos incluimos la **inversión inicial** necesaria para el **diseño y desarrollo** de la aplicación, registro mercantil de la empresa.

Dentro de lo que es el diseño de la aplicación incluiríamos los sueldos de los investigadores que se encargarían de definir e investigar las funcionalidades y especificaciones técnicas de la aplicación.

En este caso se ha considerado un grupo de 3 expertos que se dedicasen durante 2 meses a la selección y diseño de las mejores técnicas de ML/AI a implementar en la aplicación.

Costes iniciales	Cantidad (€)
Diseño de la aplicación /especificaciones técnicas	28.800,00 €
Desarrollo de la aplicación	175.000,00 €
Registro mercantil	3.000,00 €
Total	206.800,00 €

Diseño de la aplicación	Horas	Sueldo €/h	nº personas	Total
Investigación	320	30	3	28.800,00 €

Tabla 6-1: Predicción costes iniciales

- **Costes operacionales:**

Incluyen los gastos de **márquetin, mantenimiento, servicio Cloud, y soporte técnico** necesarios para el correcto funcionamiento de la aplicación.

1. **Marketing:**

La inversión en marketing será sustancialmente superior el año del lanzamiento de la aplicación e irá estabilizándose en los años siguientes.

Para el **primer año** se ha establecido un presupuesto igual al **30% del coste del desarrollo** del producto, es decir 525000€ y para los años siguientes tres irá reduciendo hasta llegar al **15 % donde se mantendrá constante**.

2. Para el **mantenimiento** se ha establecido una cantidad fija e igual al 20% del coste del desarrollo.
3. En cuanto al coste del servicio Cloud se ha establecido un gasto de 10000 euros el primer año con un incremento del 5% anual en los años siguientes.
4. En cuanto al coste asociado **soporte técnico** irá aumentando a medida que crece el volumen de clientes un 10% anual partiendo de los 20000 euros iniciales del primer año.

Costes	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
	Cantidad (€)					
Costes iniciales	206.800,00 €					
Marketing		52.500,00 €	43.750,00 €	35.000,00 €	26.250,00 €	26.250,00 €
Mantenimiento de la aplicación		43.750,00 €	43.750,00 €	43.750,00 €	43.750,00 €	43.750,00 €
Servidor Cloud		10.000,00 €	10.500,00 €	11.025,00 €	11.576,25 €	12.155,06 €
Soporte técnico		20.000,00 €	22.000,00 €	24.200,00 €	26.620,00 €	29.282,00 €
i+D constant innovation		8.750,00 €	8.750,00 €	8.750,00 €	8.750,00 €	8.750,00 €

Tabla 6-2: Proyección costes estimados

6.2.2 PROYECCIÓN INGRESOS

Los ingresos de la empresa vendrán fundamentalmente de las **comisiones** de gestión. La media de estas comisiones de otros robo-advisors se encuentran entre el 0% y el 0,75% de la cantidad invertida por los usuarios.

Para este caso particular se ha escogido una **comisión de 0,5%**.

Para la predicción de la evolución del volumen gestionado se tomará como referencia el caso de indexa capital con unas predicciones mucho más conservadores y se plantearán los siguientes tres posibles escenarios:

	Volumen bajo gestión M€					
	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
Escenario pesimista	0	3	10	25	50	90
Escenario neutro	0	6	15	43	100	140
Escenario optimista	0	10	50	100	190	300
	Ingresos €					
Escenario pesimista	0	15000	50000	125000	250000	450000
Escenario neutro	0	30000	75000	215000	500000	700000
Escenario optimista	0	50000	250000	500000	950000	1500000

Tabla 6-3: Proyección según los diferentes escenarios

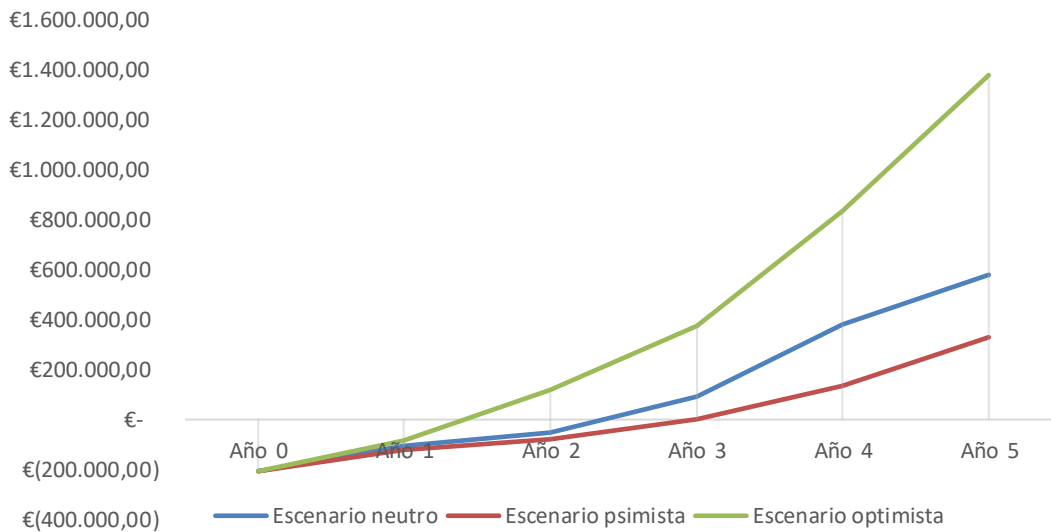


Figure 6-1: Gráfico evolución ingresos por escenarios.

Del gráfico superior se pueden extraer las siguientes conclusiones:

Escenario optimista: En este escenario en proyecto comenzarías a ser rentable en menos de un año y medio.

Escenario neutro: Según este escenario en un periodo de tiempo algo superior a los dos años el proyecto sería rentable.

Escenario pesimista: Es este caso tendríamos que esperar al tercer año para empezar a ver beneficios.

Teniendo en cuenta todo lo mencionado anteriormente, en caso de querer pasar de esta investigación académica en innovación empresarial tendría que incorporarse un socio financiero que aportase alrededor de 450.000€ para desarrollar las líneas generales identificadas y explotar la oportunidad de negocio en el mercado.

Capítulo 7. ANÁLISIS DE RESULTADOS,

CONCLUSIONES Y TRABAJOS FUTUROS

En este punto final se expondrán las **conclusiones** extraídas tras la realización del proyecto, así como las **dificultades** encontradas durante su ejecución y **próximos pasos** a seguir para trabajos futuros.

La evolución de los productos y servicios siempre va dirigida o bien hacia la **reducción de costes** o ahora más que nunca a aumentar el grado de **personalización** de estos, y dado que los RAs ya han alcanzado el objetivo de reducción de costes mediante la automatización de sus procesos, estos se irán centrando cada vez más en el aumento progresivo del grado de adaptabilidad a cada uno de los usuarios. Es por ello por lo que todas las técnicas, ejemplos y métodos que se han nombrado a lo largo del proyecto van dirigidas a conseguir ese mayor grado de personalización que el mercado busca y valora cada vez más.

Las conclusiones se van a dividir en las siguientes partes:

- Conclusiones acerca de la implementación de **la IA** para la identificación del **perfil de riesgo del cliente**.
 - Conclusiones acerca de la **IA** dentro de la **gestión de activos**.
 - Conclusiones acerca de la utilización de **sistemas de recomendación** para **productos financieros**.
1. **Conclusiones acerca de la implementación de la inteligencia artificial para la identificación del perfil de riesgo del cliente.**

Como se ha visto, el test de idoneidad es una de las partes fundamentales, no solo para los RAs sino para cualquier firma regulada que ofrezca recomendaciones de inversión.

Sin embargo, la identificación del perfil de riesgo del cliente no se trata de una tarea fácil y como se puede ver en la siguiente cita extraída de las **pautas de 2012 de la ESMA** :

“Investors, and especially retail investors, tend to underestimate the level of risk that they are taking as well as their own risk absorbing capacity. They realize that there is risk in the proposed investment, but they might not evaluate correctly the probability of the realization of the risk in certain circumstances. Risk is an overly abstract concept to govern suitability. Investors might also act in an overconfident way by considering that they are better than other investors at assessing risks and will do better. In addition, the investment firm itself, in good faith, might also underestimate the amount of risk which is being incurred by the client.”

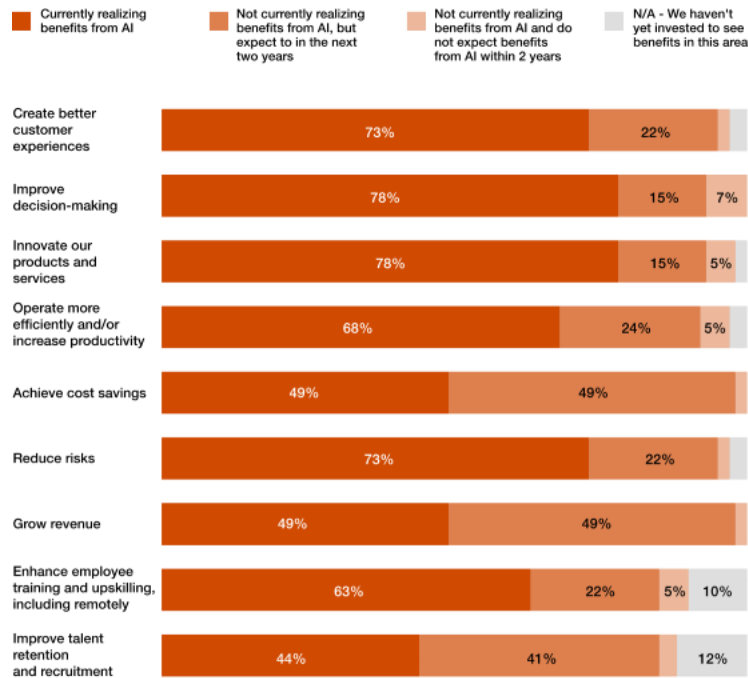
Los **inversores no suelen estimar correctamente el riesgo** que están asumiendo o el que con capaces de tolerar.

Es por ello por lo que se deben buscar nuevas formas más efectivas y confiables para la asignación del perfil de riesgo de los inversores y en este aspecto considero que la **inteligencia artificial tiene un gran potencial**, especialmente si es posible **contar con los datos** y la **información** necesaria que permitan **crear modelos fiables**.

2. Conclusiones acerca de la IA dentro de la gestión de activos.

Si hablamos de datos en cuanto al uso de IA dentro del *Asset management* vemos que la inmensa mayoría de los gestores de activos han reportado **beneficios reales** como: mejora de la toma de **decisiones**, **innovación** en los productos y servicios, mejor **experiencia** de usuario y **reducción del riesgo**.

Para mostrar con números lo mencionado, se presenta la siguiente gráfica extraída de un artículo publicado por PWC en 2021 donde se refleja los beneficios de la inteligencia artificial en la gestión de carteras.



Q3. To what degree have your AI investments in the following areas lived up to expectations? Source: PwC 2021 AI Predictions. Base: 41

Figure 7-1: Como AI ayuda a los gestores de carteras a mejorar la experiencia de los usuarios (AI for asset and wealth managers in 2021: PwC)

A pesar de que en el trabajo solo se hayan mencionado dos técnicas para la colocación de activos actualmente se están llevando a cabo múltiples estudios y cada vez surgen nuevas técnicas y modelos que pueden ayudar a establecer un reparto de capital dentro de la cartera de forma óptima.

Si analizamos más en concreto los modelos vistos en el proyecto de MVP y HRP podemos decir lo siguiente:

En general la metodología HRP evita la condición de invertibilidad de la matriz de covarianza y ofrece un enfoque eficiente que ayuda a construir carteras más robustas en términos de diversificación de pesos, minimización de riesgos y métricas de rendimiento.

Se han hecho múltiples estudios comparado estos dos sistemas obteniendo diferentes resultados dependiendo de las condiciones y activos que se hayan empleado, pero en

general se puede decir que con el uso del HRP se consiguen menores niveles de volatilidad y *drawdown* y en definitiva este ofrece mejores niveles rendimiento en términos de riesgo.

Por esta razón considero que estas nuevas técnicas innovadoras y dinámicas pueden favorecer la creación de portafolios más robustos.

3. Conclusiones acerca de la utilización de sistemas de recomendación para productos financieros.

Actualmente la cantidad de productos financieros que existen y se ofrecen a los inversores particulares es realmente grande por lo que la tarea de selección puede llegar a ser abrumadora y paralizante, especialmente para un inversor primerizo.

Por esta razón considero que los sistemas de recomendación pueden ser de gran utilidad a la hora de ayudar a los inversores en esta tarea.

Además, dentro de la inversión responsable que va a continuar con su tendencia creciente y donde cada vez existen más empresas que se encargan de evaluar a las compañías en temática ESG, junto con la creciente cantidad de normativas y regulaciones que exigen una documentación más detallada favorecerá el aumento de información y datos que podrán ser usados por los inversores o sistemas de recomendación para tomar decisiones más informadas y ajustadas a lo que los clientes buscan.

A continuación, se muestra un extracto de un artículo publicado por Clarity AI recientemente que muestra exactamente esto que se acaba de mencionar.

*“It’s been over 20 years since the founder of Apple, Steve Jobs said **“Get closer than ever to your customers. So close that you tell them what they need well before they realize it themselves”**, and yet the message is more important than ever before, especially in the world of investment.*

*The power of digitalization is bringing personalized wealth management solutions to a new segment of investors, the mass market. Often made up of **GenZ’s and Millennials**, this digitally savvy consumer segment prefers brands to be aligned with their personal values.*

As a result, offering tailored environmental, social and corporate governance (ESG) investments to support these interests represents a huge opportunity for a range of complimentary financial and non-financial platforms, including retail banks. However, without the right technology to offer the right investment product at the right time, reaching this audience is impossible.”

Analizando en concreto la utilidad de los sistemas empleados en el modelo prototipo podemos decir:

- Top 5 de los ETFs más seleccionados por los usuarios.

Se presentan los 5 ETFs más populares de entre los seleccionados por los usuarios. Es el sistema más simple y sencillo dentro de los sistemas de recomendación. A pesar de su sencillez puede ser de gran utilidad ofreciendo al usuario nuevas ideas según la opinión popular.

- Sistema basado en el contenido que emplea las **descripciones** de los ETFs para evaluar su similitud.

Como vimos este sistema emplea un análisis de la descripción de los diferentes ETFs para ofrecer aquellos que sean más similares a los que mayor puntuación haya dado el usuario.

Problema: El principal problema que presenta el algoritmo es la escasez de detalle en las descripciones que limitan la precisión del algoritmo. Sería necesario contar con una descripción más elaborada y detallada que permitiese ajustar mejor las recomendaciones.

- Sistema basado en el contenido que emplea otras **características de los ETFs** para evaluar su similitud.

Este caso es bastante similar al anterior con la diferencia de que en vez de buscar la similitud por las descripciones las busca según las características de los ETFs como pueden ser el tipo de inversión, región o la clase de activos...

Este tipo de sistemas ofrecerá recomendaciones completamente diferentes al sistema anterior y podría ser interesante siempre que el usuario tenga cierto grado de conocimientos de inversión y su intención sea buscar fondos con características específicas, sin embargo, siendo este el caso, este método podría sustituirse perfectamente por un simple filtro de características cuya implementación es mucho más sencilla y ofrece resultados similares.

- Recomendación que despliega los ETFs con mejores predicciones de puntuación basándose en la media ponderada según la similitud con otros usuarios. (Colaborative filter – Memory based – User-based).

En este caso miramos la similitud existente con otros usuarios según las puntuaciones otorgadas a los diferentes ETFs.

Los principales problemas de este sistema son como se vio en el 4.4 , son el hecho de que inicialmente no contamos con la información de muchos usuarios por lo que este método no será útil hasta que se cuente con un número suficiente de clientes, y además debido a que cada usuario únicamente puntúa 5 ETFs dentro de una base de datos de más de 25 , la matriz que se emplea en el algoritmo sigue quedando bastante vacía lo que reduce la precisión del sistema.

- Sistema Híbrido que combina un sistema basado en contenido según la similitud entre ETFs a partir de sus descripciones y un sistema que emplea SVD.
- Sistema Híbrido que combina un sistema basado en contenido según la similitud entre ETFs a partir de sus descripciones y un sistema que emplea el (Colaborative filter – Memory based – User-based) .

Los dos sistemas híbridos solventan algunos de los problemas mencionados anteriormente combinando dos modelos diferentes. Aunque siguen teniendo algunas de las carencias mencionadas con anterioridad.

En cuanto a las **conclusiones generales** finales que se han podido extraer en referencia a estos sistemas y su aplicación para este tipo de productos financieros tras la realización del proyecto son:

- En segundo lugar, al tratarse de ETFs, los cuales son un conglomerado de empresas hace que sea realmente **difícil la asignación de características concretas** ya que estas dependen de las múltiples empresas que lo forman y sus características específicas quedan diluidas.
Es por esta razón que creo que el uso de este tipo de sistemas puede ser más útil o tener mayor aplicación si estos se **enfocan a recomendaciones de compañías concretas y no tanto en Fondos o ETFs**.
- La mayor dificultad o punto en contra que se ha encontrado es la **falta de información más detallada** que permita afinar mejor las recomendaciones ofrecidas. Como se menciona el artículo del comienzo del capítulo, es necesario disponer de tecnología adecuada que ofrezca a los usuarios la información de la forma más clara y transparente posible. Poco a poco gracias a la aparición de nuevas plataformas como la que ofrece Clarity facilitarán esta tarea.
- Como se mencionó en el capítulo de sistemas de recomendación, en el caso de compras que solo se realizan una o pocas veces en la vida, como puede ser este caso o la compra de una casa, un **filtro basado en el contenido** donde el comprador introduce las características que busca y el sistema simplemente filtra aquellos productos que las contengan puede ser el sistema que más se ajuste y no requiere de ningún tipo algoritmo inteligente y a pesar de su sencillez pueden ser este tipo de recomendaciones las más adecuadas en este caso.

A modo de conclusión diría que sí es interesante emplear sistemas de recomendación que ayuden a las personas a filtrar la inmensa cantidad de productos financieros que existen simplificando así la tarea de selección y evitando la parálisis que sufren muchos inversores a la hora de enfrentarse a la inmensidad de productos que el mercado ofrece.

Sin embargo, **sistemas sencillos pueden ofrecer buenos resultados, y no es tanto el propio sistema como lo es la obtención clasificación y gestión de los datos e información lo que determinará la calidad de las recomendaciones.**

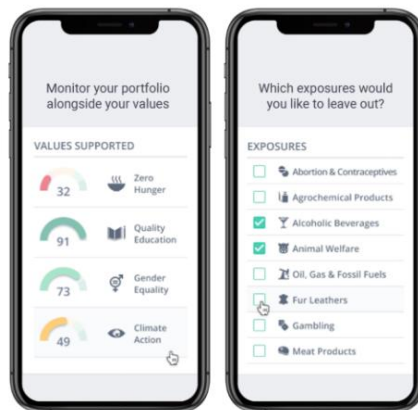


Figure 7-2: Additiv Impact Investing with Clarity AI

Por ejemplo, si se dispone de información lo suficientemente fiable y granulada de las compañías en temática ESG un buen sistema de recomendación podría basarse en un cuestionario sencillo que guíe al usuario (similar al que se ve en la imagen superior) y que sirva como filtro de sus preferencias. Si además quiere ofrecerse alguna recomendación más original podría complementarse este con un filtro colaborativo basado en el usuario utilizando por ejemplo la técnica vista del SVD.

Finalmente hablaremos de las **dificultades** presentadas y pasos a seguir en **futuros proyectos**.

En cuanto a las **dificultades**, destacan las siguientes:

- La primera complicación del proyecto es la propia **amplitud** de este. Durante este trabajo se han tratado múltiples temas que van desde la inversión responsable, robo-advisors , colocación de activos , sistemas de recomendación...

Esto ha hecho la labor de **investigación** más complicada y ha dificultado la **profundización** en alguno de los temas. Sin embargo, ha permitido ofrecer una **visión amplia** y los suficientemente detallada a cerca de estos temas tan mencionados hoy en día.

- En cuanto al aspecto más técnico del trabajo la mayor parte de las complicaciones han sido debidas a la dificultad de encontrar ETFs que ofrezcan una **información lo suficientemente detallada** como para poder ofrecer recomendaciones significativas y útiles para el inversor.
- En cuanto a la colocación de activos, dado que productos como una temática concreta, como es el caso de este proyecto que se ha centrado en productos con temática medioambiental, muestran una **fuerte correlación**, no es recomendable la construcción de carteras basadas únicamente en una temática concreta, sino que sería conveniente su combinación con **activos de otras clases y temáticas**.
- Por último, en relación con lo que se menciona en el párrafo anterior también se ha observado que el hecho de que el prototipo implementado deje a **merced del usuario la selección de los productos** de la cartera puede llevar a **problemas en cuanto a la correlación y diversificación** de los mismos resultandos en carteras con un **mayor riesgo asociado**, ya que los gustos del usuario pueden provocar que este elija productos muy parecidos y con una fuerte correlación. Sin embargo, esto no es preocupante ya que puede solucionarse fácilmente **añadiendo nuevas restricciones** o implementando estas en un **sistema de recomendación más complejo** que tenga consideraciones de este estilo en cuenta.

Finalmente, los aspectos que no han podido ser tratados o considerados en este trabajo, que hubiese sido interesantes tratar por lo que se dejan **para futuras implementaciones** son:

- Tener en cuenta las **restricciones** mencionadas a cerca de las **selecciones de los activos por parte de los usuarios**, para que estos se ajusten al perfil de riesgo del usuario teniendo en cuenta **la ratio entre renta variable y renta fija**, así como la **correlación** de los activos que formen parte de la cartera.
- Lanzar la página web y contratar un servicio de **alojamiento en la nube** para almacenar los datos de los usuarios y a partir de los resultados evaluar la efectividad y rendimiento de los diferentes sistemas de recomendación, realizando un estudio comparativo entre ellos.
- **Análisis y estudio de dichos datos** para ver si existen características demográficas que puedan influir en las preferencias de los usuarios como pueden ser el género, localización geográfica o edad.
- Empleo de información de los **datos** obtenidos de usuarios para **mejorar, analizar e identificar** los algoritmos de **recomendación** con mejores resultados.
- Propuesta de varios métodos basados en *machine learning* para la asignación del **perfil de riesgo y comparación de sus resultados** con el modelo tradicional del test de idoneidad.

Capítulo 8. ALINEACIÓN CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE

Si analizamos la alineación de este proyecto con los Objetivos de desarrollo sostenible podemos ver que tiene bastante impacto en varios de ellos, más bien se podría decir que tiene relación e impacto en todos ellos ya que lo que se trata de alguna manera es de promover y fomentar la inversión en empresas que estén alineados con estos objetivos de desarrollo.

Si tuviésemos que destacar algunos haría especial referencia a los siguientes ya que los fondos que se ofrecen en el prototipo de robo-advisor se centran en las siguientes temáticas:

- **7. Energía asequible y no contaminante:** Garantizar el acceso a una energía asequible, segura, sostenible y moderna.
- **11.Ciudades y comunidades sostenibles:** Lograr que las ciudades sean más inclusivas, seguras, resilientes y sostenibles.
- **10.Reducción de las aguas residuales:** Reducir la desigualdad en y entre los países.
- **13.Acción por el clima:** Adoptar medidas urgentes para combatir el cambio climático y sus efectos.
- **8. Trabajo decente y crecimiento económico:** Promover el crecimiento económico inclusivo y sostenible, el empleo y el trabajo decente para todos.

Capítulo 9. BIBLIOGRAFÍA

- [29] Banik, R. (2018). *Hands-On Recommendation Systems with Python: Start building powerful and personalized, recommendation engines with Python (English Edition) (1.a ed.)*. Packt Publishing.
- [30] Boreiko, D. (2020). *How Risk Profiles of Investors Affect Robo-Advised Portfolios*. Frontiers. Recuperado 24 de mayo de 2022, de <https://www.frontiersin.org/articles/10.3389/frai.2020.00060/full>
- [31] C. (2020, 4 diciembre). *Guía para entender los diferentes enfoques de inversión de la ESG y poder clasificar los fondos*. FundsPeople España. Recuperado 24 de mayo de 2022, de <https://fundspeople.com/es/guia-para-entender-los-diferentes-enfoques-de-inversion-de-la-esg-y-poder-clasificar-los-fondos/>
- [32] Finanbest. (2022, 18 mayo). *Home*. Gestor automatizado de inversión | Líder en rentabilidad. Recuperado 24 de mayo de 2022, de <https://www.finanbest.com/>
- [33] *Goals Archive*. (2022, 15 marzo). The Global Goals. Recuperado 24 de mayo de 2022, de <https://www.globalgoals.org/goals/>
- [34] Kane, F. (2021). *Building Recommender Systems with Machine Learning and AI: Help people discover new products and content with deep learning, neural networks, and machine learning recommendations*. Independently published.
- [35] Lewinson, E. (2022). *Python for Finance Cookbook: Over 60 powerful recipes for effective financial data analysis, 2nd Edition*. Packt Publishing - ebooks Account.
- [36] N. (2021, 13 octubre). *Turning client suitability into a science*. Neuroprofiler. Recuperado 24 de mayo de 2022, de <https://neuroprofiler.com/en/client-suitability/>
- [37] PricewaterhouseCoopers. (2021). *AI for asset and wealth managers*. PwC. Recuperado 24 de mayo de 2022, de <https://www.pwc.com/us/en/tech-effect/ai-analytics/ai-predictions/asset-and-wealth-management.html>
- [38] S. (2022, 24 mayo). *How to Develop a FinTech App in 2022 A Guide to Action*. SpdLoad. Recuperado 7 de junio de 2022, de <https://spdload.com/blog/fintech-app-development/>
- [39] Sironi, P. (2016). *Fintech Innovation: From Robo-Advisors to Goal Based Investing and Gamification (1. ed.)*. Wiley.

- [40] Tatsat, H., Puri, S., & Lookabaugh, B. (2020). *Machine Learning and Data Science Blueprints for Finance: From Building Trading Strategies to Robo-Advisors Using Python*. O'Reilly Media.
- [41] M. Day, T. Cheng and J. Li, "AI Robo-Advisor with Big Data Analytics for Financial Services," 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2018, pp. 1027-1031, doi: 10.1109/ASONAM.2018.8508854.
- [42] Zibriczky12, D. (2016). Recommender systems meet finance: a literature review. In Proc. 2nd Int. Workshop Personalization Recommender Syst (pp. 1-10).

ANEXO I

Características EFTs empleados:

TFname	Name	Descrip
TAN	Invesco Solar ETF	The cleverly-named TAN delivers targeted exposure to the solar power energy, making it potentially useful for both betting on long-term adoption of this energy source or capitalizing on perceived short-term mispricings. Like many granular ETFs focusing on specific sub-sectors, TAN doesn't offer tremendous diversification; there are only about 35 individual components—including both U.S. and international stocks—with three or four names accounting for a third of assets. Investors seeking broad-based exposure to clean energy may want to take a look at ICLN or GEX, while KWT offers another option for targeted solar power exposure. TAN's hyper-targeted focus makes it appropriate only for a select few, but for those seeing to overweight the solar power space this ETF can be a nice option.
QCLN	First Trust NASDAQ Clean Edge Green Energy Index Fund	QCLN is a unique member of the Alternative Energy Category, as this ETF invests in companies that are engaged in a variety of different activities related to several green energy sub-sectors. By including companies focused on biofuels, solar energy, and advanced batteries (among others), QCLN casts a wide net of exposure. That, along with a relatively deep basket of individual companies, may make it appealing for those seeking broad-based exposure to alternative energy sources. There are cheaper options for exposure to alternative energy, but few that offer the depth and breadth of holdings QCLN can boast.

<p>XLK</p>	<p>Technology Select Sector SPDR Fund</p>	<p>State Street’s XLK grants investors the opportunity to gain exposure to numerous powerhouse tech firms all under a single ticker. To be included in this ETF, a company must be one of numerous sectors under the technology umbrella. This includes market segments like IT services, wireless telecommunication services, and semiconductors to name just a few. The fund invests in the who’s-who of the U.S. tech sector, with major holdings in companies like Apple and IBM. The fund splits its assets mainly between the technology and communication services sectors, while allocating mainly to giant and large cap firms. One of the major strengths of this ETF is the fact that it does not single out a particular sector; rather it invests in companies from all across the technology sector. This makes the fund an ideal choice for investors who want tech exposure, but are unsure as to which particular segment of this broad market that they feel will perform the best.</p>
<p>NULG</p>	<p>Nuveen ESG Large-Cap Growth ETF</p>	<p>NULG invests in companies within the MSCI USA Growth Index (the parent index) which exhibit positive ESG characteristics. The growth aspect is determined by five metrics: long- and short-term forward EPS growth, current internal growth rate, long-term forward EPS, and long-term historical sales growth. The index then ranks companies within the parent index according to their impact on climate change, natural resource use, waste and emission management, safety, ethics, and compliance with national and international rules and regulations, among other categories. Companies with significant activity involving weapons, gambling, alcohol, and tobacco are specifically excluded. The top 50% of these ESG-ranked growth companies are selected for inclusion, per sector within the index and then weighted by a multi-factor optimizer that aims to keep holdings and performance similar to the parent index.</p>
<p>AQWA</p>	<p>Global X Clean Water ETF</p>	<p>AQWA provides exposure to the global water industry by investing in companies of all capitalization that derive at least 50% of their revenue from either water infrastructure or water equipment and/or services. The fund commences by screening public information using a proprietary natural language processing algorithm to identify firms that are defined as clean water companies and are involved in the treatment, purification, conservation and management of water. Eligible securities must then meet minimum ESG requirements as defined by the UN Global Compact principles to be included in the index</p>

<p>CGW</p>	<p>Invesco S&P Global Water Index ETF</p>	<p>This ETF offers exposure to a group of companies operating generally in the water industry, including both water utilities, infrastructure companies, and water equipment and materials companies. As such, this ETF likely doesn't belong in a long-term buy-and-hold portfolio due to the targeted nature of exposure, but may be appealing to those who believe that scarcity issues will prompt increased demand for water treatment companies. While this investment thesis may seem compelling, it is not clear how strong the link between water usage/scarcity trends and performance will be going forward. Given the complexity of the issues, as well as the various other business operations of component companies, we're skeptical of the ability of this ETF to accomplish the objective investors may be expecting of it. Moreover, CGW faces some concentration issues, as a few individual stocks receive huge allocations, and it is on the pricey side.</p>
<p>EBLU</p>	<p>Ecofin Global Water ESG Fund</p>	<p>EBLU offers exposure to developed-market water companies of all capitalization sizes. The fund selects firms that derive at least 40% of gross revenues from either water infrastructure or water equipment and/or services. Water infrastructure companies are those that provide public water distribution, engineering, construction or consulting. Water equipment companies are those who provide water pipes, valves, pumps or water efficiency products such as filtration, treatment, and testing of water. Service companies provide technologies that facilitate management of water distribution, usage, treatment, and irrigation. Components must also meet minimum ESG requirements to be included in the index. The resulting portfolio includes a few dozen names, heavily weighted towards the US</p>

PHO	Invesco Water Resources ETF	<p>This ETF offers exposure to a group of companies operating generally in the water industry, including both water utilities and infrastructure companies and water equipment and materials companies. As such, this ETF likely doesn't belong in a long-term buy-and-hold portfolio due to the targeted nature of exposure, but may be appealing to those who believe that scarcity issues will prompt increased demand for water treatment companies. While this investment thesis may seem compelling, it is not clear how strong the link between water usage/scarcity trends and performance will be going forward. Given the complexity of the issues, as well as the various other business operations of component companies, we're skeptical of the ability of this ETF to accomplish the objective investors may be expecting of it. Moreover, PHO faces some concentration issues, as a few individual stocks receive huge allocations, and is on the pricey side, however, this fund is the cheapest in the category</p>
PIO	Invesco Global Water ETF	<p>This ETF offers exposure to a group of companies operating generally in the water industry, including both water utilities and infrastructure companies and water equipment and materials companies. As such, this ETF likely doesn't belong in a long-term buy-and-hold portfolio due to the targeted nature of exposure, but may be appealing to those who believe that scarcity issues will prompt increased demand for water treatment companies. While this investment thesis may seem compelling, it is not clear how strong the link between water usage/scarcity trends and performance will be going forward. Given the complexity of the issues, as well as the various other business operations of component companies, we're skeptical of the ability of this ETF to accomplish the objective investors may be expecting of it. Moreover, PIO faces severe concentration issues, as a few individual stocks receive huge allocations— arguably the most so out of the Category— and it is the priciest fund in the Category as well.</p>

<p>OCEN</p>	<p>IQ Clean Oceans ETF</p>	<p>OCEN is passively managed to hold an ESG portfolio of companies engaged in ocean-related sustainability. The portfolio aims to cast a wider net than just water sustainability. The index selects from a universe of companies involved in circular economy, clean energy, water and waste treatment, electronic components, energy efficient economy, seafood and alternatives, shipping and ports, and sustainable packaging. Developed and emerging market-specific screens are initially applied to screen out companies that fail to meet ESG criteria based on global sustainability trends and stakeholder management. Companies that are non-compliant with the UN Global Compact or are engaged in controversial activities are also excluded. Eligible companies are each given a thematic score based on revenue generated from thematic exposure and impact on the environment.</p>
<p>SDG</p>	<p>iShares MSCI Global Sustainable Development Goals ETF</p>	<p>SDG tracks an ESG-focused index composed of global companies that derive at least 50% of their revenues from products and services that further the United Nation's Sustainable Development Goals. These goals focus on health and diet, sustainable energy, and education. Companies are explicitly deselected if more than 10% of sales are from alcohol, tobacco, or are involved in predatory lending. The index must contain at least 30 issuers, with an issuer cap of 4% and a sector cap of 20%. Constituent weight is based on the percent of income derived from the above-stated impact sales, scaled by trailing 12-month revenues, and rebalanced annually. Given the fund's narrow construction and lose constraints, the fund could vary considerably from cap weighted exposure and performance, and may not be comparable with broadbased indices.</p>

<p>ERTH</p>	<p>Invesco MSCI Sustainable Future ETF</p>	<p>The Invesco MSCI Sustainable Future ETF invests in companies from around the world that offer products and services that contribute to a more environmentally sustainable economy. The fund invests in small-, mid- and large-cap companies from around the world. Companies are eligible for inclusion in the index if they derive 75% or more of their cumulative revenue from six areas: alternative energy, energy efficiency, green building, sustainable water, pollution prevention and control, and sustainable agriculture. The index excludes companies that faced very severe controversies related to environmental, social and governance issues in the last three years, as well as companies involved in controversial weapons. The fund is among dozens of ETFs that target companies that compare favorably on environmental, social and governance criteria, also known as ESG. ESG funds are an increasingly popular segment of the ETF marketplace, offering values-driven investors a diverse portfolio of U.S. stocks without compromising their conscience. ERTH is part of a narrower subset of ESG known as impact funds, whose goal is to invest in companies that try to bring about a measurable, beneficial social or environmental impact. Invesco’s fund fees are reasonable for the segment, though fees for impact ETFs tend to be significantly higher than plain-vanilla index funds and some broad-based ESG funds. The holdings are also significantly narrower. Due to the increased diversification and concentration risk of its portfolio, ERTH is not a good replacement for a core global equity position though it may be a good complement for investors committed to sustainable businesses.</p>
<p>WWOW</p>	<p>Direxion World Without Waste ETF</p>	<p>WWOW provides concentrated exposure to all-cap, US-listed companies and ADRs considered as representatives of the transformative economy shift from linear (one-user/one-use) to a circular (multi-owner/re-use) model. The selection universe begins with the eligible firms that derive 50% or more of their revenue from the five sub-themes: sustainability of resources (renewable energy, biofuels), sharing platforms (crowdfunding, content sharing), resource recovery (waste management), product as a service (subscription economy), and life cycle extension (reselling and trading). The top 10 companies with the largest market capitalization from each sub-theme are selected for the index. If less than 10 securities qualify in a sub-theme, the fund may select from the remaining list of eligible firms by market-cap, until the portfolio contains a total of 50 equities</p>

<p>NUMV</p>	<p>Nuveen ESG Mid-Cap Value ETF</p>	<p>NUMV aims to invest in the most socially responsible companies while still providing reasonably market-like exposure to midcap value stocks. The index takes its holdings from the MSCI USA Mid-Cap Value Index (the parent index), which selects value stocks according to price/book, forward price/earnings, and dividend yield. From there, NUMV screens out companies involved in controversial businesses, such as alcohol, weapons, nuclear power, and gambling. Companies that exceed carbon emissions thresholds are also excluded. The remaining firms are scored on ESG factors, and the highest scoring firms within each sector (top 50% by market cap) are selected for inclusion in the index. The portfolio is weighted according to a multi-factor optimization algorithm that aims to keep holdings and performance similar to the parent index</p>
<p>PBD</p>	<p>Invesco Global Clean Energy ETF</p>	<p>This ETF offers exposure to the global clean energy index, including both U.S. and international stocks in the underlying portfolio. PBD also diversifies across various types of clean energy—such as wind, solar, and hydro—making it an interesting option for those looking to bet on a clean energy boom but unwilling to make a concentrated bet on a specific sub-sector. Relative to competing options such as ICLN or GEX, PBD may be appealing because it maintains a much smaller concentration of assets among its top holdings—thereby minimizing the fund’s dependence on a few select stocks. On the downside, PBD is significantly more expensive than some alternatives, particularly ICLN. Those who value balanced exposure may find that PBD is worth the extra fees charged relative to other options.</p>
<p>XHB</p>	<p>SPDR S&P Homebuilders ETF</p>	<p>This ETF is focused on the U.S. homebuilding industry, and as such offers exposure to a corner of the domestic economy that tends to be cyclical in nature. In addition to pure play homebuilders, this fund includes companies related generally to the homebuilding industry, such as Pier One. For investors seeking exposure to the homebuilding industry—or the closest thing to it available in an ETF wrapper—we think XHB is the best option out there. This fund is more cost efficient than other options such as PKB or ITB, and the equal weighting methodology ensures exposure is spread evenly across component companies.</p>

SUSA	iShares MSCI USA ESG Select ETF	SUSA offers socially conscious exposure to US large- and midcap equities. It holds MSCI USA companies with the highest ESG ratings, determined by the degree to which they reflect socially conscious values with regard to the environment, society, and corporate governance. As such, companies with tobacco and civilian firearms involvement as well as those with low MSCI ESG controversies score are excluded from the index. SUSA weights its holdings in proportion to their ESG scores. At the same time, the fund caps security weights at 5% and limits sector deviation from the broader MSCI USA index to 3%, ensuring that the fund doesn't deviate far from the equity market. Index constituents are reviewed on a quarterly basis
IQSI	IQ Candriam ESG International Equity ETF	IQSI is an index-tracking fund designed to deliver exposure to international developed companies meeting environmental, social and corporate governance criteria developed by IndexIQ and Candriam. To form the initial index universe, the fund combines sector-specific analysis and ranking of issuer-specific factors, like corporate governance and labor practices, along with a macro analysis of the issuers exposure to global sustainability trends, including health and wellness. Companies in the top 70% within each sector are selected. Companies are further screened based on any continued and significant non-compliance to Candriams ESG criteria to exclude from the final index
SDGA	Impact Shares Sustainable Development Goals Global ETF	SDGA assesses the universe of global large- and midcap stocks according to social investment criteria developed by the UN Capital Development Fund. These 32 metrics dealing with company policies and practices regarding business ethics, employment practices, contractor & supply chain monitoring, community involvement, and financial inclusion are used to create a societal development score for each eligible firm. In addition, firms must have a minimum amount of business exposure to UN-designated Least Developed Countries. SDGA excludes companies involved in serious business controversies or certain controversial industries, or that do not meet a minimum ESG score.

ESGN	Columbia Sustainable International Equity Income ETF	ESGN consists of large- and mid-cap stocks pulled from the MSCI World ex USA Index, excluding REITs, selected and weighted according to their dividend characteristics. In order to be eligible for inclusion in the fund, each company must have a minimum MSCI ESG Rating of BB and a 1% dividend yield. The minimum ESG (environmental, social, and corporate governance) rating indicates that all companies in the fund have positive exposure to ESG issues. Eligible securities are scored based on their dividend yield, dividend growth, and cash-based dividend coverage. Dividend growth is the ratio of the difference between the current and previous dividend per share by the previous annual dividend per share. Cash-based dividend coverage measures free cash flows against their current annualized dividend
ESGD	iShares ESG Aware MSCI EAFE ETF	ESGD aims to increase exposure to developed market firms, excluding the US and Canada, displaying strong ESG traits while exhibiting risk and return characteristics similar to its parent index through an optimization process. Companies in the broad MSCI EAFE Index are rated on risk factors related to environmental (e.g. carbon emissions, water use, toxic waste), social (e.g. labor management, health & safety, sourcing), and governance issues (e.g., corruption, fraud, anticompetitive practices). Additionally, the fund completely excludes tobacco companies, producers of certain weapons (e.g. landmines, bioweapons), and companies experiencing severe business controversies.
EASG	Xtrackers MSCI EAFE ESG Leaders Equity ETF	The Xtrackers MSCI EAFE ESG Leaders Equity ETF (EASG) tracks an MSCI index of developed market stocks outside North America, selecting those securities that score the highest relative to their peers on environmental, social and governance factors, known by the acronym ESG. EASG includes companies that score in the top 50% of scores in each sector — and so will own about half as many companies as the parent index — then weights those stocks to keep the sector allocation in line with the parent index. The fund excludes companies involved in alcohol, tobacco, gambling, controversial and conventional weapons, nuclear power, and civilian firearms. EASG’s management fee is a bit cheaper than rival iShares ESG MSCI EAFE ETF (ESGD), but ESGD had a head start and has significantly greater assets and daily liquidity. The iShares ESGD follows a less stringent MSCI ESG index for the region and so has more stocks in its portfolio than Xtrackers EASG. The idea is that companies with higher ESG scores will outperform their rivals over the long run.

RESD	WisdomTree International ESG Fund	RESD actively selects international stocks by screening for a mix of fundamental and technical factors, including quality, value, momentum and correlation. The fund also scores securities for sustainability favoring positive ESG attributes and excluding companies involved in controversial activities, such as tobacco, small arms, controversial weapons and fossil fuels. Holdings are weighted on a modified market-cap basis, giving heavier weight to constituents with higher market-cap and positive sustainability scores
VEGI	iShares MSCI Global Agriculture Producers ETF	This ETF gives investors indirect exposure to agricultural commodity prices through an international portfolio of equities engaged in the agribusiness industry. VEGI's underlying holdings are poised to benefit from the ongoing increase in food demand from developed and emerging markets alike, and furthermore it can double as a hedge since agricultural commodities are often among the first to rise in inflationary environments. Investors looking to tap into this corner of the market will likely find VEGI as an appealing instrument. First and foremost, this ETF offers by far the most diverse portfolio of holdings amongst agriculture ETFs. Additionally, VEGI features the lowest expense ratio among agriculture ETFs,
MOO	VanEck Agribusiness ETF	This ETF gives investors worldwide exposure to a basket of equities that are involved in the agriculture business. MOO's holdings are primarily in developed regions, although the fund still has emerging market exposure in Brazil, Singapore, and Malaysia. The fund's holdings are poised to benefit from the ongoing increase in food demand from developed and emerging markets alike, and furthermore it doubles as a hedge since agricultural commodities are often among the first to rise in inflationary environments.

<p>SCRD</p>	<p>Janus Henderson Sustainable Corporate Bond ETF</p>	<p>SCRD is an actively managed fund that primarily invests in USD denominated investment grade debt, issued by companies the adviser believes exhibit sustainable business practices. The funds universe is first constructed by excluding issuers that dont comply with the UN Global Compact, and/or that derive a majority of their revenue from industries that negatively impact the society or environment. Using proprietary criteria, securities are selected that align with certain positive social and environmental themes. These themes are: 1) Transition to a Green Economy, 2) Economic and Community Development and Inclusion, 3) Knowledge & Technology, and Innovation, and 4) Health & Well-Being. A fundamental research process is also utilized, which considers third party data, analysis and ratings to construct the final portfolio. SCRD may allocate a small portion of the fund to invest in high-yield securities, US Treasurys, emerging market loans, and cash.</p>
<p>VCSH</p>	<p>Vanguard Short-Term Corporate Bond ETF</p>	<p>VCSH offers exposure to investment grade corporate bonds that fall towards the short end of the maturity spectrum, thereby delivering a moderate amount of credit risk but limiting exposure to rising interest rates. Like most Vanguard ETFs, VCSH is among the most cost-efficient in its ETFdb Category. VCSH might be useful for investors looking to enhance fixed income returns through additional credit risk but also interested in shortening up effective duration.</p>
<p>BECO</p>	<p>BlackRock Future Climate and Sustainable Economy ETF</p>	<p>BECO is actively managed to hold globally-listed companies of any market-cap that are believed to help further the transition to a lower carbon economy, including those involved in sustainable energy, circular economy, and the future of transport and nutrition. Eligible companies are assessed for proprietary environmental criteria such as metrics related to carbon intensity and carbon emissions. BECO seeks to align with the goals of the Paris Climate Agreement, which aims for net-zero greenhouse gas emissions by 2050. The fund also includes sector-specific alignment targets for select sectors and does not intend to invest in certain higher emission industries and in companies violating the UN Global Compact. Additionally, the fund attempts to have a better aggregate environmental assessment and overall ESG score than its benchmark, the MSCI ACWI Multiple Industries Select Index.</p>

<p>SUSC</p>	<p>iShares ESG Aware USD Corporate Bond ETF</p>	<p>SUSC is positioned to be a building block for investors looking to structure their own ESG focused portfolio. The fund tracks a corporate bond index that considers each issuer`s environmental, social, and governance practices. Companies involved in tobacco or controversial weapons are excluded. The fund also avoids firms subject to severe business controversies (such as anti-competitive, environmental, or health and safety issues) in the past three years. The remaining bonds are rated according to a quantitative ESG rating methodology developed by MSCI ESG Research. The ratings consider key ESG-related risks and opportunities by industry, such as climate change, resource scarcity, or demographic shifts. Final selection and weighting are done by an optimizer, which maximizes exposure to positive ESG traits while retaining market-like risk and return.</p>
<p>BGRN</p>	<p>iShares Global Green Bond ETF</p>	<p>BGRN targets bonds that align with MSCIs Green Bond Principles. Qualifying bonds must be issued to fund projects in one of six categories: alternative energy, energy efficiency, pollution prevention and control, sustainable water, green building, and climate adaption. Additionally, bond issuers must maintain processes for evaluating and selecting green projects, managing proceeds, and reporting environmental impact. BGRN may still hold bonds that dont meet all the criteria if they were issued before 2014 and have been widely accepted as green, as determined by the index committee. BGRNs selection pool includes both government and corporate issuers from developed and emerging markets. Holdings may be of any term and are generally held to maturity.</p>
<p>WFIG</p>	<p>WisdomTree U.S. Corporate Bond Fund</p>	<p>WFIG's eligible universe begins with investment grade, US-domiciled, USD denominated debt, with no maturity mandate. From here, the fund selects its fixed income securities by scoring companies based on their free cash flows over debt service, leverage ratios, and return on invested capital, and subsequently removing the lowest-scoring 20%. The fund weights its securities by scoring remaining bonds by their probability of default and duration, giving preference to riskadjusted higher income. The fundamental approach intends to identify companies with healthy balance sheets and cash flows, while tilting towards higher income.</p>

EAGG	iShares ESG Aware US Aggregate Bond ETF	EAGG puts the aggregate bond market through an ESG filter. All US Treasury, MBS, CMBS, and ABS exposure is preserved at market weight, as these securities are viewed as ESG-neutral. Remaining constituents (corporate bonds and government-related bonds) are excluded if the entities are involved in tobacco, controversial weapons, or civilian firearms, as well as entities involved in very severe business controversies. The remaining bonds are then scored based on general ESG criteria, such as pollution, anti-competitive practices, and health and safety. The final portfolio is optimized to maximize the average ESG score, while preserving marketlike risk and return characteristics. EAGG focuses on investment-grade bonds of various maturities
NUHY	Nuveen ESG High Yield Corporate Bond ETF	NUHY filters the components of the Bloomberg MSCI U.S. High Yield Very Liquid Index (the base index) for issues that satisfy certain ESG criteria such as climate change, natural resource use, company relations, safety and business ethics. The index generally excludes companies with significant activities in controversial businesses such as alcohol, tobacco, and firearms. Eligible securities are then market-value-weighted within each sector, with sector weights adjusted to mirror the base index. ESG ratings are generally updated annually but can be more frequent in the index providers discretion.
HYXF	iShares ESG Advanced High Yield Corporate Bond ETF	HYXF is passively-managed to invest in high yield, USD-denominated corporate bonds issued by developed market countries. Index construction begins with the Bloomberg US Corporate High Yield Index (parent index), it includes bonds that have USD 400 million or more of outstanding face value. Using MSCI ESG research, companies rated poorly for ESG controversy score are excluded. In addition, the underlying index excludes companies involved in fossil fuel-related activities and other activities such as: genetically modified organisms, alcohol, gambling, nuclear weapons, civilian firearms, palm oil, for profit prisons, and predatory lending. Firms that pass the screenings are then rated based on 37 key ESG factors to select those with a high average rating.

EUSB	iShares ESG Advanced Total USD Bond Market ETF	<p>EUSB is passively-managed to provide exposure to US dollar denominated taxable bonds from issuers with high ESG ratings and avoid involvement in serious business controversies. The parent index includes a variety of: Treasury securities, corporate bonds, and mortgage-backed securities. EUSB excludes issuers with low ESG ratings, according to MSCI ESG Research, and securities of issuers involved in activities such as: alcohol, gambling, genetic engineering, nuclear weapons, civilian firearms, private prisons, and predatory lending. Fossil fuel companies are excluded by removing any company in the Bloomberg energy sector and all companies with a tie to thermal coal, oil and gas. MSCI identifies key ESG issues for each industry, annually. Each security Issuer is scored based on the risk exposure and risk management occurring within the last three years. Issuers are rated and ranked in comparison to their industry peers</p>
------	--	--

ETFname	Asset Class	Asset Class Size	Strategy	Region	ESG Score	ESG Score Global Percentile (%)	Revenue Exposure to Environmental Impact (%)	Revenue Exposure to Social Impact (%)	Severe Governance Controversies (%)
TAN	Equity	Multi-Cap	Fundamental	Developed Markets	7,72	0,6803	0,6193	0	0
QCLN	Equity	Multi-Cap	Vanilla	North america	7,14	0,5274	0,408	0	0
XLK	Equity	Large-cap	Vanilla	North america	10	1	0,0875	0,0019	0,0002
NULG	Equity	Large-cap	Growth	North america	8,45	0,8345	0,0664	0,0612	0,0256

AQWA	Equity		ESG	Global	10	1	0,2367	0,0211	0
CGW	Equity	Multi-Cap	Equal	Developed Markets	8,88	0,8963	0,208	0,0208	0
EBLU	Equity	Multi-Cap	ESG	Developed Markets	9,7	0,9705	0,1831	0,0351	0
PHO	Equity	Multi-Cap	Vanilla	North america	10	1	0,1462	0,0224	0
PIO	Equity	Multi-Cap	Vanilla	Developed Markets	9,73	0,9717	0,1355	0,0652	0
OCEN	Equity		ESG		8,95	0,9056	0,1279	0,0495	0,0885
SDG	Equity	Large-cap	ESG	Developed Markets	8,03	0,7516	0,4039	0,3372	0

ERTH	Equity	Multi-Cap	ESG	Developed Markets	7,55	0,6315	0,695	0,0014	0,0298
WWOW	Equity		Equal	North america	7,47	0,6113	0,1944	0	0
NUMV	Equity	mid-cap	Value	North america	8,44	0,8327	0,0487	0,0297	0
PBD	Equity	Multi-Cap	Equal	Developed Markets	6,7	0,4367	0,344	0	0,0147
XHB	Equity	mid-cap	Equal	North america	7,39	0,592	0,0587	0,0477	0
SUSA	Equity	Large-cap	ESG	North america	8,9	0,899	0,0473	0,0351	0,0085
IQSI	Equity	Large-cap	ESG	North america	9,37	0,949	0,0388	0,0495	0,0985

ICAI ICADE CIHS

SDGA	Equity	Large-cap	ESG	Developed Markets	9,29	0,9428	0,0529	0,0374	0,1836
ESGN	Equity	Multi-Cap	Fundamental	Developed Markets	9,73	0,971	0,0501	0,0363	0,0085
ESGD	Equity	Large-cap	ESG	Developed Markets	9,42	0,9517	0,0434	0,0496	0,069
EASG	Equity	Large-cap	ESG	Developed Markets	9,45	0,9531	0,0397	0,051	0,0489
RESD	Equity	Large-cap	Active	Developed Markets	9,24	0,9377	0,0288	0,0694	0,0647
VEGI	Equity	Multi-Cap	Vanilla	Developed Markets	7,84	0,7068	0,0736	0,0758	0
MOO	Equity	Large-cap	Vanilla	Developed Markets	8,84	0,8909	0,056	0,0652	0

SCRD			active	North america	8,04	0,754	0,0605	0,0231	0,0568
VCSH					9,11	0,9252	0,0183	0,0116	0,0211
BECO	Equity		Active	Global	9	0,9144	0,1537	0,0253	0,0337
SUSC	Bond		ESG	Developed Markets	8,6	0,857	0,0282	0,0205	0,1188
BGRN	Bond		ESG	Developed Markets	7,86	0,7109	0,0569	0,0123	0,0541
WFIG					7,71	0,6762	0,0193	0,0142	0,22
EAGG	Bond		ESG	North America	7,17	0,5341	0,0089	0,0058	0,0374

ICAI ICADE CIHS

NUHY	Bond		ESG	North America	7,68	0,6687	0,0229	0,0298	0,0365
HYXF	Bond		ESG		6,9	0,4725	0,0209	0,0307	0
EUSB	Bond		ESG	North America	7,43	0,6	0,0109	0,012	0,012