John Soldatos
Dimosthenis Kyriazis *Editors*

Big Data and Artificial Intelligence in Digital Finance

Increasing Personalization and Trust in Digital Finance using Big Data and AI

OPEN ACCESS

Springer

Big Data and Artificial Intelligence in Digital Finance

John Soldatos • Dimosthenis Kyriazis
Editors

Big Data and Artificial Intelligence in Digital Finance

Increasing Personalization and Trust in Digital Finance using Big Data and AI

Springer

Editors
John Soldatos
INNOV-ACTS LIMITED
Nicosia, Cyprus

University of Glasgow
Glasgow, UK

Dimosthenis Kyriazis
University of Piraeus
Piraeus, Greece

ISBN 978-3-030-94589-3 ISBN 978-3-030-94590-9 (eBook)
https://doi.org/10.1007/978-3-030-94590-9

© The Editor(s) (if applicable) and The Author(s) 2022. This book is an open access publication.
Open Access This book is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.
The images or other third party material in this book are included in the book's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Part III Applications of Big Data and AI in Digital Finance

Part IV Applications of Big Data and AI in Insurance

Chapter 10
Next-Generation Personalized Investment Recommendations

Richard McCreadie, Konstantinos Perakis, Maanasa Srikrishna, Nikolaos Droukas, Stamatis Pitsios, Georgia Prokopaki, Eleni Perdikouri, Craig Macdonald, and Iadh Ounis

1 Introduction to Investment Recommendation

In developed nations, there is growing concern that not enough people are saving for their later life and/or retirement. For instance, 35% (18.4 million) of the UK adult population reported they do not have a pension, and 43% admit they do not know how much they will need for retirement according to a study by Finder.[1] One of the most stable ways to save effectively is through long-term investment in a broad portfolio of financial assets [13]. However, for the average citizen, investing is a difficult, time consuming and risky proposition, leading to few doing so. Hence, there is significant interest in technologies that can help lower the barriers to entry to investment for the average citizen.

Meanwhile, recent advances in Big Data and AI technologies have made the idea of an automated personal assistant that can analyse financial markets and recommend sound investments seem possible. Indeed, there are already a range of online services that will provide financial investment recommendations for a fee.

[1] https://www.finder.com/uk/pension-statistics.

R. McCreadie (✉) · M. Srikrishna · C. Macdonald · I. Ounis
University of Glasgow, Glasgow, Scotland
e-mail: richard.mccreadie@glasgow.ac.uk; maanasa.srikrishna@glasgow.ac.uk; craig.macdonald@glasgow.ac.uk; iadh.ounis@glasgow.ac.uk

K. Perakis · S. Pitsios
UBITECH, Chalandri, Greece
e-mail: kperakis@ubitech.eu; spitsios@ubitech.eu

N. Droukas · G. Prokopaki · E. Perdikouri
National Bank of Greece, Athens, Greece
e-mail: nikolaos.droukas@nbg.gr; georgia.prokopaki@nbg.gr; eleni.perdikouri@nbg.gr

© The Author(s) 2022

J. Soldatos, D. Kyriazis (eds.), *Big Data and Artificial Intelligence in Digital Finance*,
https://doi.org/10.1007/978-3-030-94590-9_10

Manual financial investment advice services such as Seeking Alpha[2] and Zacks Investment Research[3] have existed for over a decade, while new automated 'Robo-Advisors' like SoFi Automated Investing[4] and Ellevest[5] have been gaining traction over the last few years. The primary advantages of such automatic systems are that they are more accessible and scalable than other forms of investment advice, while also being flexible with regard to how 'hands-on' the investor wants to be with their portfolio [35]. Hence, such systems are seen as one viable solution to providing personalized financial recommendations for the public.

The European Commission's H2020 INFINITECH project is investigating these technologies within its 'Automated, Personalized, Investment Recommendations System for Retail Customers' pilot, led by the National Bank of Greece. The primary use-case within this pilot is producing automated financial asset recommendations for use by the bank's financial advisors, either during a physical meeting between advisor and customer within a bank branch or remotely. The goals here are threefold: (i) enhanced productivity (i.e. improved productivity of investment consultants of the bank, through faster access to recommendations tailored to their retail customer accounts), (ii) better advice for investments based on a better understanding of customer preferences and behaviour and (iii) increased trading volumes, by widening the pool of investors the bank services.

In the remainder of this chapter, we will discuss the whys and hows of developing an automated financial asset recommender for this use-case. In particular, the chapter is structured as follows. In Sect. 2, we begin by describing the regulatory environment for such systems in Europe, to provide some context of where regulatory boundaries exist that need to be considered. Section 3 then formally introduces the definition of what a financial asset recommender needs to perform in terms of its inputs and outputs, while Sect. 4 discusses how to prepare and curate the financial data used by such systems. In Sect. 5, we provide a review of the scientific literature that is relevant to creating different types of financial asset recommendation systems. Additionally, in Sect. 6, we present experimental results produced within the scope of the INFINITECH project, demonstrating the performance of different recommendation systems using past customers from the National Bank of Greece and over the Greek stock market. A summary of our conclusions is provided in Sect. 7.

[2] https://seekingalpha.com/.

[3] https://www.zacks.com/.

[4] https://www.sofi.com.

[5] https://www.ellevest.com/.

2 Understanding the Regulatory Environment

Before diving into the practicalities of developing a financial asset recommendation system, it is important to consider the regulatory environment that such a system needs to exist within. Within Europe, there are two main pieces of legislation of interest, which we discuss below:

General Data Protection Regulation (GDPR) GDPR is an EU regulation that was introduced in 2018 and governs the use of personal data regarding EU citizens and those living within the European Economic Area (EEA), and however it has subsequently been used as the basis for similar laws in other countries and so is more widely applicable. GDPR is relevant to the development of financial recommendation systems, as such systems by their nature will need to process the personal data of each customer to function. The key provisions in GDPR to consider therefore are:

- *Data Residency*: All personal data must be stored and remain securely within the EU, and there must be a data protection officer responsible for that data.
- *Pseudonymization*: Personal data being processed must be sufficiently anonymized such that it cannot be subsequently attributed to a specific person without the use of further data.
- *Data Access and Correction*: The customer must be able to access the data stored about them and be able to update that data on request.
- *Transparent Processing*: The customer needs to be made aware of how their data will be used and explicitly accept the processing of their data.

Markets in Financial Instruments Directive (MiFID) MiFID has been applicable across the European Union since November 2007 and forms the cornerstone of the EU's regulation of financial markets, seeking to improve their competitiveness by creating a single market for investment services and activities and to ensure a high degree of harmonized protection for investors in financial instruments. Later in 2011, the European Commission also adopted a legislative proposal for the revision of MiFID, resulting in the Directive on Markets in Financial Instruments repealing Directive 2004/39/EC and the Regulation on Markets in Financial Instruments, commonly referred to as MiFID II and MiFIR, which are currently in effect. These regulations are quite wide-ranging, as they cover the conduct of business and organizational requirements for investment firms, authorization requirements for regulated markets, regulatory reporting to avoid market abuse, trade transparency obligations for shares and rules on the admission of financial instruments to trading.

The MiFID Directive regulates all investment products, including Equities, Bonds, Mutual Funds, Derivatives, Structured Deposits (but not other products such as Loans or Insurance). From the perspective of these regulations, there are three key provisions to account for when developing financial asset recommendation systems (note that if the system is also performing buy/sell orders, there are other provisions in addition to the following to consider):

- *Investors should be classified based on their investment knowledge and experience*: This means that any financial asset recommendation system needs to factor in the investment history of the customer and provide information that is appropriate to their level of expertise.
- *Suitability and appropriateness of investment services need to be assessed*: From a practical perspective, not all financial assets or vehicles will be suitable for all types of investors, this provision means that the types of assets recommended must be filtered to only those that are appropriate for the situation of the customer.
- *Fully inform investors about commissions*: If a commission fee is charged by the underlying services that are being recommended, these must be clearly displayed.

3 Formalizing Financial Asset Recommendation

From a customer's perspective, the goal of financial asset recommendation is to provide a ranked list of assets that would be suitable for the customer to invest in, given the state of the financial market and the constraints and desires of that customer. In this case, 'suitability' is a complex concept that needs to capture both the profitability of the asset with respect to the market and the likelihood that the customer would actually choose to invest in that asset (requiring solutions to model factors such as the perceived risk of an asset in combination with the customer's risk appetite). The notion of suitability also has a temporal component, i.e. how long the customer wants to invest their capital for (known as the investment horizon). With regard to what precisely is recommended, these might be individual financial assets, but more likely are a group of assets (since it is critical to mitigate risk exposure by spreading investments over a broad portfolio). Despite this, from a system perspective, it is often more convenient to initially treat financial asset recommendation as a scoring problem for individual assets, i.e. we identify how suitable an asset is in isolation first, and then tackle aggregation of individual assets into suitable asset groups as a second step. Hence, individual asset scoring can be formulated as follows:

Definition 1 (Individual Asset Scoring) Given a financial asset 'a' (e.g. a stock or currency that can be traded) from a set of assets within a financial market $a \in \mathcal{M}$, a customer 'c' with a profile $\mathcal{P}_c$ and an investment time horizon $\mathcal{T}$ produce a suitability score $s(\mathcal{H}_a, \mathcal{P}_c, \mathcal{M}, \mathcal{T})$, where $0 \leq s \leq 1$ and a higher score indicates that a is more suitable for investment. Here $\mathcal{H}_a$ represents the historical properties of the asset a, $\mathcal{P}_c$ represents the customer profile, $\mathcal{M}$ comprises contextual data about the market as a whole and $\mathcal{T}$ is the investment time horizon (e.g. 3 years).

Asset History $\mathcal{H}_a$ When considering an asset's history $\mathcal{H}_a$, we typically consider the past pricing data for that asset on a daily basis (such as open/close prices, as well as highs/lows for that asset each day). Such time series data can be converted into multiple numerical features describing the asset across different time periods (e.g. the last [1,3,6,12,24,36] months). For instance, we might calculate the Sharpe Ratio over each period to provide an overall measure of the quality of the asset or use volatility to measure how stable the gains or losses from an asset are (representing risk).

Customer Profile $\mathcal{P}_c$ To enable personalization of the recommendations for a particular customer, it is important to factor in the financial position and preferences defined by their profile $\mathcal{P}_c$. The most important factors to consider here are the liquidity of the customer (i.e. how much they can invest safely) and the customer's risk appetite (which should influence the weighting of associated high/low risk assets). We may also have other historical data about a customer in their profile as well, such as past financial investments, although this is rare as most customers will not have invested in the past.

Factoring in the Market $\mathcal{M}$ Markets will behave differently depending on the types of assets that they trade, which is important since we should consider profitability in the context of the market. In particular, there are two ways that we can consider profitability: (1) raw profitability, typically represented by annualized return on investment (where we want returns upward of 7% under normal conditions) and (2) relative profitability, which instead measures to what extent an investment is 'beating' the average of the market (e.g. by comparing asset return to the market average). Relative profitability is important to consider as markets tend to follow boom/bust cycles, and hence a drop in raw profitability might reflect the market as a whole rather than just the asset currently being evaluated.

Investment Time Horizons $\mathcal{T}$ Lastly, we need to consider the amount of time a customer has to invest in the market, as this can strongly impact the suitability of an asset. For example, if working with a short horizon, then it may be advantageous to avoid very volatile assets that could lose the customer significant value in the short term unless they are particularly risk seeking.

By developing an asset scoring function $s(\mathcal{H}_a, \mathcal{P}_c, \mathcal{M}, \mathcal{T})$, we can thereby use the resultant scores to identify particularly good assets to either recommend to the customer directly or include within a broader portfolio of assets. The same scoring function can similarly be used to update asset suitability (and hence portfolio quality) over time as more data on each asset $\mathcal{H}_a$ and the market as a whole $\mathcal{M}$ is collected over time. Having formulated the task and introduced the types of data that can be used as input, in the next section, we discuss how to prepare and curate that data.

4 Data Preparation and Curation

4.1 Why Is Data Quality Important?

Data quality issues can have a significant impact on business operations, especially when it comes to the decision-making processes within organizations [5]. As a matter of fact, efficient, accurate business decisions can only be made with clean, high-quality data. One of the key principles of data analytics is that the quality of the analysis strongly depends upon the quality of the information analysed. However, according to Gartner,[6] it is estimated that more than 25% of critical data in the world's top companies are flawed, at the same time when data scientists worldwide spend around 80% of their time on preparing and managing data for analysis, with approximately 60% of their time being spent on cleaning and organizing data and with an additional 19% of their time being spent on data collection.[7] It is thus obvious that in our big data era, actions that can improve the quality of diverse high volume (financial) datasets are critical and that these actions need to be facilitated by (automated, to the maximum extent possible) tools, optimizing them in terms of (time) efficiency and effectiveness.

The complete set of methodological and technological data management actions for rectifying data quality issues and maximizing the usability of the data are referred to as *data curation* [7]. Data curation is the active and on-going management of data through its lifecycle of interest and usefulness, from creation and initial storage to the time when it is archived for future research and analysis, or becomes obsolete and is deleted [4]. Curation activities enable data discovery and retrieval, maintaining quality, adding value and providing for re-use over time. Data curation has emerged as a key data management activity, as the number of data sources and platforms for data generation has grown. *Data preparation* is a sub-domain of data curation that focuses on data pre-processing steps, such as aggregation, cleaning and often anonymization. Data preparation is the process of cleaning and transforming raw data prior to processing and analysis. It often involves reformatting data, making corrections to that data and combining datasets to add value. The goal of data preparation is the same as other data hygiene processes: to ensure that data is consistent and of high quality. Inconsistent low-quality data can contribute to incorrect or misleading business intelligence. Indeed, it can create errors and make analytics and data mining slow and unreliable. By preparing data for analysis up front, organizations can be sure they are maximizing the intelligence potential of that information. When data is of excellent quality, it can be easily processed and analysed, leading to insights that help the organization make better decisions. High-

[6] https://www.reutersevents.com/pharma/uncategorised/gartner-says-more-25-percent-critical-data-large-corporations-flawed.

[7] https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#34d3b5ad6f63.

quality data is essential to business intelligence efforts and other types of data analytics, as well as better overall operational efficiency.

4.2 Data Preparation Principles

To make financial data useable, that data must be cleansed, formatted and transformed into information digestible by the analytics tools that follow in a business intelligence pipeline. The actual data preparation process can include a wide range of steps, such as consolidating/separating fields and columns, changing formats, deleting unnecessary or junk data and making corrections to that data.[8] We summarize each of the key data preparation steps below:

Discovering and Accessing Data Data discovery is associated with finding the data that are best-suited for a specific purpose. Data discovery can be a very painful, frustrating and time-consuming exercise. An essential enabler of efficient discovery is the creation and maintenance of a comprehensive, well-documented data catalogue (i.e. a metadata repository). In this context, key data sources include asset pricing data, company and customer profiles, as well as investment transaction data.

Profiling and Assessing Data Profiling data is associated with getting to know the data and understanding what has to be done before the data becomes useful in a particular context and is thus key to unlocking a better understanding of the data. It provides high-level statistics about the data's quality (such as row counts, column data types, min, max and median column values and null counts), and visualization tools are very often exploited by users during this phase, enabling them to profile and browse their data. This is particularly important in the finance context to identify and understand unusual periods in financial time series data, e.g. caused by significant real-world events like COVID-19.

Cleaning and Validating Data Cleaning up the data is traditionally the most time-consuming part of the data preparation process, but it is crucial for removing faulty data and filling in any data gaps found. Some of the key data cleansing activities include:

- Making corrections to the data, e.g. correcting timestamps with a known amount of lag
- Deleting unnecessary data, e.g. deleting extra or unnecessary fields used across different sources in order to cleanly consolidate the discrete datasets
- Removing outliers, e.g. replacing outliers with the nearest 'good' data or with the mean or median, as opposed to truncating them completely to avoid missing data points

[8] https://blogs.oracle.com/analytics/what-is-data-preparation-and-why-is-it-important.

- Filling in missing values, i.e. predicting and imputing data missing at random or not at random
- Otherwise, conforming data to a standardized pattern

Once data has been cleansed, it must be validated by testing for errors in the data preparation process up to this point. Often times, an error in the system will become apparent during this step and will need to be resolved before moving forward. Outlier removal is particularly important here, as individual data points such as very high or low prices (e.g. caused by an erroneous trade or a malicious actor) can confuse machine learning systems.

Transforming Data Data transformation is the process of changing the format, structure or values of data. Data transformation may be constructive (adding, copying and replicating data), destructive (deleting fields and records), aesthetic (standardizing salutations or street names) or structural (renaming, moving and combining columns in a database). Proper data transformation facilitates compatibility between applications, systems and types of data.

Anonymizing Data Data anonymization is the process of protecting private or sensitive information by masking, erasing or encrypting identifiers (e.g. names, social security numbers and addresses) that connect an individual to stored data. Several data anonymization techniques exist, including data masking, pseudonymization, generalization, data swapping, data perturbation and more, each one serving different purposes, each coming with its pros and cons and each being better suited than the others depending upon the dataset at hand, but all serving the common purpose of protecting the private or confidential information included within the raw data, as required by law (see Sect. 2).

Enriching Data Data enrichment is a general term that applies to the process of enhancing, refining and improving raw data. It is the process of combining first-party data from internal sources with disparate data from other internal systems or third-party data from external sources. As such, data enrichment could be interpreted both as a sub-step of the data cleansing process, in terms of filling in missing values, and as an independent step of the process. This may require the association (or linking) of the raw data with data from external sources or the enrichment of the raw data with meaningful metadata, thus facilitating their future discovery and association and thus empowering the extraction of deeper insights. A common example here is connecting asset pricing data with profiles for associated companies that influence those assets.

Storing Data Once prepared, the data can be stored or channelled into a third-party application—such as a business intelligence tool—clearing the way for processing and analysis to take place.

4.3 The INFINITECH Way Towards Data Preparation

Within the context of the INFINITECH project, the consortium partners have designed and developed a complete data ingestion/data preparation pipeline, aiming to provide an abstract and holistic mechanism for the data providers, addressing the various connectivity and communication challenges with the variety of data sources that are exploited in the finance and insurance sector, as well as the unique features that need to be considered when utilizing a range of heterogeneous data sources from different sectors. Hence, the scope of the Data Ingestion mechanism is fourfold:

1. To enable the acquisition and retrieval of heterogeneous data from diverse data sources and data providers
2. To facilitate the mapping of the entities included in the data to the corresponding entities of an underlying data model towards its annotation (metadata enrichment)
3. To enable the data cleaning operations that will address the data quality issues of the acquired data
4. To enable the data anonymization operations addressing the constraints imposed by GDPR and other related national and/or European (sensitive) data protection regulations and directives

The INFINITECH complete data ingestion/data preparation pipeline is composed of four main modules, as also illustrated in Fig. 10.1, depicting the high-level architecture of the pipeline. We describe each of these modules in more detail below:

Data Retrieval This module undertakes the responsibility to retrieve or receive the new datasets from a data source or data provider either periodically or on-demand. The scope of the Data Retrieval module is to facilitate the data acquisition from any relational database, HDFS deployments, FTP or HTTP servers, MinIO storage servers, as well as from any API of the data source. The prerequisite is that the appropriate information is provided by the data provider. Additionally, the Data Retrieval module enables the ingestion of new datasets that are pushed from the data provider to its exposed RESTful APIs. Hence, the Data Retrieval module supports all the aforementioned data source types which are considered the most commonly used data sources in current Big Data ecosystems. Nevertheless, the modular architecture of the described solution facilitates the future expansion of the list of supported data source types in an effortless and effective manner.

Data Mapper This module is responsible for the generation of the mapping between the entities of a retrieved dataset and the ones of an underlying data model (derived from the data provider's input). The scope of the Data Mapper module is to enable the mapping of the data entities from a new dataset to the data model that is given by the data provider. In this sense, the data provider is able to create the mappings for each entity of the new dataset to a specific data entity of the data model. To achieve this, at first the Data Mapper module offers the means to

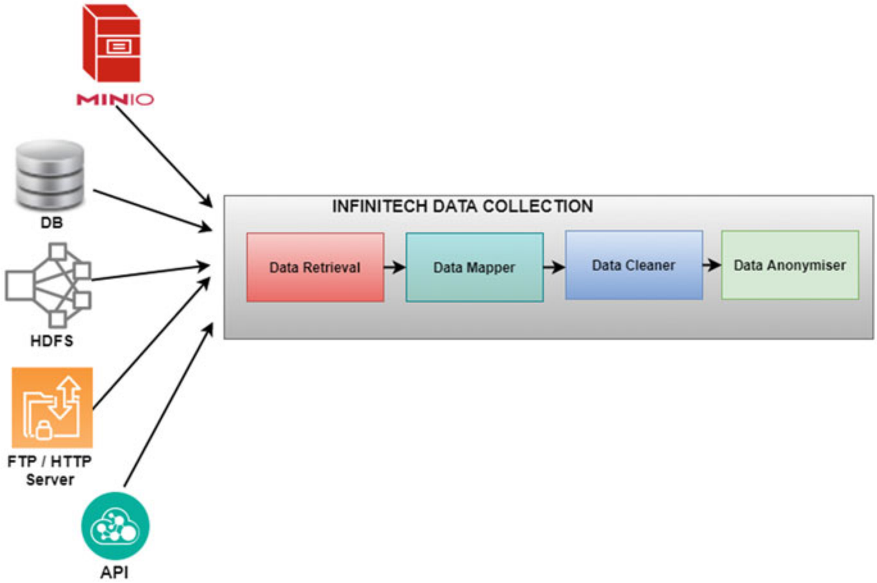

Fig. 10.1 INFINITECH data preparation pipeline

integrate a data model during its initial configuration. Then, during processing, the data entities of the provided dataset are extracted and displayed to the data provider via its user friendly and easy-to-use user interface. Through this user interface, the data provider is able to select the corresponding entities of the integrated data model that will be mapped to the entities of the dataset. The generated mappings are stored for later reuse in a JSON format.

Data Cleaner This module undertakes the responsibility to perform the data cleaning operations on the retrieved dataset, based on the data provider's input. The scope of the Data Cleaner is to provide the data cleaning operations that will ensure that the provided input datasets, which originate from a variety of heterogeneous data sources, are clean and complete to the maximum extent possible. The specific functionalities of this module enable the detection and correction (or removal) of inaccurate, corrupted or incomplete values in the data entities, with the aim of increasing data quality and value. To this end, the data cleaning process is based on a set of data cleansing rules that are defined by the data provider on a data entity level via the Data Cleaner's user friendly and easy-to-use user interface and is a four-step process that includes:

1. The validation of the values of the data entities against a set of constraints
2. The correction of the errors identified based on a set of data correction operations
3. The data completion of the values for the required/mandatory data entities with missing values with a set of data completion operations

4. The maintenance of complete history records containing the history of errors identified and the data cleaning operations that were performed to address them

Data Anonymizer This module undertakes the responsibility of addressing the various privacy concerns and legal limitations imposed on a new dataset as instructed by the data provider. To this end, the anonymizer provides the advanced privacy and anonymization toolset with various data anonymization techniques that can be tailored by the data provider in order to filter or eliminate the sensitive information based on his/her needs. To achieve this, the anonymizer employs a generic data anonymization process which is highly customizable through the definition of anonymization rules which are set by the data provider leveraging his/her expertise.

All four processes can be performed as background processes, provided that the data providers have created in advance the corresponding data retrieval/mapping/cleaning/anonymization profiles for a specific dataset, which will in turn be used in an automated way for the execution of the processes. In the next section, we will discuss approaches to use our curated data to perform financial asset recommendation.

5 Approaches to Investment Recommendation

With the underlying data needed to perform asset scoring prepared, we next need to understand how our asset scoring function can be built. In this section, we will provide a brief background into one class of methods for developing such a scoring function, namely using recommendation models, with a particular focus on recommendation as it applies to financial products and services.

What Is a Recommender? Recommendation systems are a popular class of algorithms that aim to produce personalized item suggestions to a user. Popular examples of such algorithms are movie recommendation (e.g. on Netflix) [3] or product recommendation (e.g. Amazon recommended products) [36]. For our later reported results, we will only be considering supervised machine-learned recommenders, i.e. approaches that leverage machine learning to analyse the past history of items, user interactions with those items and/or the user's relationships to the items, with the goal of producing a recommendation model [15]. However, we will discuss a couple of unsupervised approaches later in this section. The process of creating a supervised machine learned model is known as training. A trained recommendation model can be considered as a function that (at minimum) takes a user and item as input and produces a score representing the strength of relationship between that user and item pair. An effective recommendation model should find a right balance between relevance, diversity and serendipity in the recommended items, using the different types of evidence discussed in Sect. 3. Hence, for financial asset recommendation, we want to train a model to represent the function $s(\mathcal{H}_a, \mathcal{P}_c, \mathcal{M}, \mathcal{T})$ (see Sect. 3), where $\mathcal{H}_a$ represents the item (a financial asset) and $\mathcal{P}_c$ represents the user (a customer).

Types of Recommender Recommendation algorithms can be divided along three main dimensions:

1. The types of evidence they base the recommendations upon.
2. Whether items as scored in isolation or as part of a set or sequence of items.
3. To what extent the recommender understands/accounts for temporal dynamics during the training process.

In terms of evidence, a model might utilize explicit interactions between users and items (e.g. when a user watches a movie) [18], intrinsic information about an item (e.g. what actors star in a movie) [41], time series data about an item (e.g. popularity of a movie over time) [37], intrinsic information about a user (e.g. the user says they like action movies) [24] or time series data about the user (e.g. the user has been watching a lot of dramas recently) [43]. When considering recommendation context, most approaches consider each item in isolation, but some approaches examine items within the context of a set (e.g. recommending an item to add to a customer's basket that already contains items [27]) or when recommending a sequence of items (e.g. recommending a series of places to visit when on holiday [19]). Finally, depending on how the model is trained, simple approaches consider all prior interactions as an unordered set, while more advanced techniques will factor in that recent interactions are likely more relevant than older interactions [40].

Within the context of the financial asset recommendation, explicit interactions between users and items are somewhat rare and difficult to obtain, as they represent past investments made by customers (which is not public data). Hence, approaches that can more effectively leverage available intrinsic and historical data about the customers and assets for recommendation are more desirable here. With regard to recommendation context, we will be focusing on asset recommendations in isolation here, and however we would direct the reader to the literature on basket recommenders if interested in recommendation of items to add to an existing portfolio [1]. Meanwhile, capturing the temporal dynamics of our historical data and interactions is intuitively important within financial markets, as past asset performance is not always reflective of future performance, and hence models that can capture temporal trends will be more effective in this domain.

In the remainder of this section, we will discuss a range of recommendation approaches that can be applied for financial asset recommendation, as well as highlight their advantages and disadvantages. To structure our discussion, we will group models into six broad types. We will start with four supervised approaches that we will experiment with later in Sect. 6, namely *Collaborative filtering*, *User Similarity*, *Key Performance Indicator Predictors*, and *Hybrid*. We then discuss two unsupervised approaches, namely *Knowledge Based* and *Association Rule Mining*.

5.1 Collaborative Filtering Recommenders

Collaborative filtering recommenders focus on providing item recommendations to users based on an assessment of what other users, judged to be similar to them, had positively interacted with in the past. These models work with a matrix of user–item interactions, which is typically sparse, owing to the fact that most customers would not have interacted with (invested in) more than a small subset of items (financial assets) previously. The intuition behind these models is that missing ratings for items can be predicted, or imputed, as they are likely correlated across the user and item space. By learning the similarities between users and items, inferences can be made about the missing values. As this approach leverages past interactions between the user and items in order to provide recommendations, it typically works best with large quantities of interactions such that re-occurring patterns can be found. These interactions can be in the form of explicit feedback, wherein a user provides an explicit, quantitative rating for the item, or as implicit feedback, where user interactions are gauged from their behaviour using system, such as clicking a link, or spending a significant amount of time browsing a specific product [1].

Advantages and Disadvantages Collaborative filtering models can be advantageous since they tend to generate diverse recommendations, since they are looking for items interacted with (invested in) by customers that are similar to you, but you have not interacted with. On the other hand, particularly when neighbourhood-based, they suffer from the problem of sparsity, where it is difficult to provide recommendations for a user who has little to no interaction data available, as correlations cannot easily be drawn. More advanced model-based techniques alleviate the sparsity problem, but often still require large amounts of existing interaction data, which presents a particular problem in the financial domain due to the difficulty of curating and obtaining sufficient anonymized financial transactions [14]. Also of note is that many of these models employ dimensionality reduction (to speed up computation and remove noise) and learn *latent* factors between users and items in order to perform recommendation. This makes explaining why certain recommendations were made challenging, as the latent factors learned may not be interpretable by humans.

Applications in Finance Collaborative filtering techniques are often applied for financial recommendation. Some notable approaches include the implementation of a fairness-aware recommender system for microfinance, by Lee et al. [16], which uses item-based regularization and matrix factorization, as well as risk-hedged venture capital investment recommendation using probabilistic matrix factorization, as proposed by Zhao et al. [45]. Luef et al. [17] used user-based collaborative filtering for the recommendation of early stage enterprises to angel investors, as a method of leveraging the investors' circle of trust. Meanwhile, Swezey and Charron [38] use collaborative filtering recommendations for stocks and portfolios, reranked by Modern Portfolio Theory (MPT) scores, which incorporate metrics of risk, returns and user risk aversion. From these approaches, it can be gleaned that while there is value in utilizing user–product interaction data to provide financial recommendation, this is not sufficient as a representation of the suitability of the recommended financial products, and augmentation with asset intrinsic information is often necessary. Furthermore, gathering the required interaction data for collaborative filtering recommendation can present a significant challenge due to the rarity of investment activity among the general populace.

5.2 User Similarity Models

Like collaborative filtering models, user similarity models are based on the intuition that if two users are similar, then similar items should be relevant to both. However, instead of directly learning latent similarity patterns between users and items from past interaction data, user similarity models generate a feature vector for each user and directly calculate similarity between all pairs of users (e.g. via cosine similarity). Once a top-n list of most similar users is found, a ranked list of items can be constructed based on items those similar users have previously interacted with (e.g. invested in), where the more users interact with an item, the higher its rank.

Advantages and Disadvantages The primary advantage of user similarity models is that they are highly explainable (e.g. 'x users like you invested in this asset'). However, in a similar way to collaborative filtering models, they are reliant on sufficient prior interaction data to identify 'good' items for groups of similar users.

Applications in Finance Luef et al. [17] implement a social-based recommender model for investing in early stage enterprises, which asks users to specify an inner circle of trusted investors and then provide recommendations on the basis of what those users invested in. Yujun et al. [44] propose a fuzzy-based stock recommender algorithm, which recommends instruments that were chosen by other similar users. Such models that focus on similar users tend to be seen as more convincing by end users, as the results can be presented in not only an explainable but also a personable manner by alluding to other experts in the field who made similar decisions.

5.3 Key Performance Indicator Predictors

KPI predictor-based recommenders utilize descriptive features of the users and items in order to produce recommendations, rather than relying on the existence of explicit or implicit interactions between those users and items. These features are typically expressed as a single numerical vector representation per user–item pair. Using these vectors as a base, content-based recommenders will typically train a model to predict a rating, score or label (KPI) that can act as a proxy for item suitability to the user within the context of the target domain. For example, in the financial asset recommendation context, such a model might aim to predict the profitability of the asset given the user's investment horizon.

Advantages and Disadvantages The advantage that KPI predictors hold over collaborative filtering is that they do not rely on past interaction data, and hence new items that have not been interacted with in the past can be recommended based on their descriptive features. Such models can also be applied to cold-start users (those with no investment history). The drawback of these models is that they rely on a proxy KPI rather than a true measure of suitability, meaning that performance will be highly dependent on how correlated the KPI is with actual suitability. Additionally, these models tend to only have a limited capacity to personalize the recommendations for each individual user, i.e. the item features tend to dominate the user features in such models.

Applications in Finance Seo et al. [33] produced a multi-agent stock recommender that conducts textual analysis on financial news to recommend stocks. This is one of several models that utilize natural language processing techniques to extract stock predictors from textual data [31, 32]. Musto et al. also develop a case-based portfolio selection mechanism that relies on user metadata for personalization [22], while Ginevicius et al. attempt to characterize the utility score of real estate using complex proportional evaluation of multiple criteria [8].

5.4 Hybrid Recommenders

In practical applications, multiple recommender models are often combined or used in tandem to improve overall performance. Such combined approaches are known as hybrid recommenders. There are two broad strategies for combining recommenders: unification and voting. Unification refers to the combination of recommender models to produce a singular algorithm that returns a result. This might involve feeding the output of multiple existing models into a weighted function that combines the scores from each (where that function may itself be a trained model). Meanwhile, voting approaches consider the output of the different learned models as votes for each item, where the items can be ranked by the number of votes they receive. Recommenders can also be cascaded, wherein multiple recommenders are given strict orders of priority. The absence of relevant information for one recommender will then trigger recommendations from the model of lower priority. This can present a useful solution to the problem of datasets being sparse or the unavailability of consistent feedback from users across various methods.

Advantages and Disadvantages The main advantage of hybrid approaches is that they alleviate the disadvantages of each individual approach and are able to recommend a diverse range of items. On the other hand, they add significant complexity to the recommendation task (particularly when additional model(s) need to be trained to perform the combination) and make outcome explainability more challenging [14].

Applications in Finance Taghavi et al. [39] proposed the development of an agent-based recommender system that combines the results of collaborative filtering together with a content-based model that incorporates investor preferences and socioeconomic conditions. Luef et al. [17] also employed a hybrid recommendation approach that compares the rankings produced by their previously described social user-based recommender and knowledge-based recommender using Kendall's correlation. Matsatsinis and Manarolis [20] conduct the equity fund recommendation task using a combination of collaborative filtering and multi-criteria decision analysis and the associated generation of a utility score. Mitra et al. [21] combine a collaborative filtering approach along with an attribute-based recommendation model in the insurance domain, in order to account for the sparsity of user ratings. Hybrid techniques can provide recommendations that are representative of the different variables that influence the suitability of financial assets for investment, such as investor preferences, textual analysis, expertly defined constraints, prior investment history and asset historical performance and perhaps present the best theoretical solution for the current problem setting. However, discovering the optimal combination of these models can be challenging, especially as their numbers increase.

5.5 *Knowledge-Based Recommenders*

Having discussed supervised approaches to the problem of financial asset recommendation, to complete our review, we also highlight a couple of unsupervised approaches, starting with knowledge-based recommenders.

Knowledge-based recommendation is based on the idea of an expert encoding a set of ‘rules’ for selecting items for a user. In this case, the system designer defines a set of filtering and/or scoring rules. At run-time, a user fills in a questionnaire on their constraints and preferences (this information might also be derived from an existing customer profile), needed to evaluate those rules. The rules are then applied to first filter the item set, then score and hence rank the remaining items, forming the recommended set [6].

Advantages and Disadvantages In terms of advantages, they allow for highly customizable and explainable recommendations, as they are based on human-defined rules. They are also usable for cold-start users with no prior history (as long as they correctly fill in the questionnaire). On the other hand, they are reliant on humans defining good generalizable rules, which is both time consuming and error-prone for diverse domains like finance and needs to be continually updated manually [6].

Applications in Finance Gonzalez-Carrasco et al. [9] utilized a fuzzy approach to classify investors and investment portfolios, which they then match along with one another. Set membership for investments is calculated using a set of predefined rules. For instance, a product in prime real estate with normal volatility might be classified as socially moderate and psychologically conservative, but an investment in solar farms with a low long-term return is classified as socially and psychologically aggressive. Luef et al. [17] evaluated the performance of multiple recommender models to recommend early stage enterprises to investors, and one such model was knowledge-based, where investor profiles are used to determine constraints on recommendation, with profile features acting as hard filters and investor preferences acting as soft constraints on enterprise profiles. These recommenders allow for the fine-grained specification of requirements and constraints upon the recommendation and are helpful in providing context to support investment decisions by presenting an integrated picture of how multiple variables influence investment instruments. These recommenders can provide highly customized results that can be tweaked as the user desires alongside their changing preferences; however, they cannot glean latent patterns representative of the userbase and their investment trends, which might prove useful in providing recommendations that leverage a user's circle of trust.

5.6 Association Rule Mining

Association rule mining is an unsupervised, non-personalized method of recommendation which seeks to identify frequent itemsets—which are groups of items that typically co-occur in baskets. In doing so, it is often used as a method of recommending items that users are likely to interact with given their prior interactions with other items identified within those itemsets. These relationships between the items, or variables, in a dataset are termed association rules and may contain two or more items [2]. This technique is typically applied to shopping settings where customers place multi-item orders in baskets and therefore is also termed market basket analysis.

Advantages and Disadvantages Association rules are straightforward to generate in that they do not require significant amounts of data or processing time, and as an unsupervised technique, the input data does not need to be labelled. However, as an exhaustive algorithm, it discovers all possible rules for the itemsets that satisfy the required thresholds of co-occurrence. Many of these rules may not be useful or actionable; however, this requires expert knowledge to evaluate and provide insight upon the meaningfulness of the generated rules.

Applications in Finance One such approach by Nair et al. [23] uses temporal association rule mining to account for the general lack of consideration of time information in typical association rule mining approaches. They leverage a genetic algorithm called Symbolic Aggregate approXimation (SAX) to represent time series data in a symbolic manner and then feed these representations into the model to obtain association rules and provide recommendations on the basis of these rules. Other approaches by Paranjape-Voditel and Deshpande [25], as well as Ho et al. [11], look at the usage of fuzzy association rule mining to investigate hidden rules and relationships that connect one or more stock indices. The former recommends stocks, by using the principle of fuzzy membership in order to calculate relevance of various items in the portfolio, and also introduces a time lag to the inclusion of real-time stock price information, so that any emerging patterns in price movement are aptly captured. This becomes especially applicable in the financial domain, wherein quantitative data about instruments can be classified better using a fuzzy approach as opposed to one using crisp boundaries, so as to identify the ranges of the parameters.

6 Investment Recommendation within INFINITECH

Having summarized how we can generate recommendation scores for assets using different types of recommendation algorithms, in this section, we illustrate how these models perform in practice. In particular, we will first describe the setup for evaluating the quality of financial asset recommendation systems and then report performances on a real dataset of financial transactions data from the Greek stock market.

6.1 Experimental Setup

Dataset To evaluate the quality of a financial asset recommendation system, we need a dataset comprising financial assets and customers, as well as their interactions. To this end, we will be using a private dataset provided by the National Bank of Greece, which contains investment transactions spanning 2018–2021, in addition to supplementary information on demographic and non-investment behaviour data from a wider subset of the bank's retail customers, and historical asset pricing data. From this dataset, we can extract the following main types of information:

- **Customer Investments**: This forms the primary type of evidence used by collaborative filtering recommendation models, as it contains the list of investment transactions conducted by all customers of the bank. These timestamped transactions contain anonymized customer IDs, instrument IDs and ISINs, as well information on the instrument names and the volume and direction of the transactions.
- **Customer Intrinsic and Historical Features**: This supplementary data includes aggregated indicators derived from the retail transactions and balances of all the anonymized customers. Such indicators include aggregated credit card

Table 10.1 Greek stock market sample dataset statistics by year

Dataset information		2018	2019	2020	2021
Market data	Unique assets	4087	4328	4883	3928
	Price data points	655,406	796,288	839,637	145,885
	Average annualized return	−6.74%	11.65%	19.22%	6.71%
	Assets involved in investment	1089	983	903	636
Customers	Customers	52,365	52,365	52,365	52,365
	Customers with investment transactions	18,359	15,925	17,060	8126
	Total transactions	72,020	73,246	121,703	47,501

spends across multiple sectors and deposit account balances with aggregations spanning different time windows and demographic features on the customers. These KPIs are recorded in entirety across 2018, 2019 and 2020, with only incomplete aggregates available across 2021. Only a limited subset of users possess investment transactions.

- **Asset Intrinsic and Historical Features**: This data contains the historical prices of the assets spanning 2018–2021, alongside other descriptive information on the assets such as their name, ISIN and financial sector. The price data points are available for almost all days (although there are a few gaps due to data collection failures, see Table 10.1).

Models For our experiments, we will experiment with eight representative supervised recommendation models from those described earlier, starting with two collaborative filtering models:

- **Matrix Factorization (MF) [Collaborative Filtering]** [28]: This is the conventional matrix factorization model, which can be optimized by the Bayesian personalized ranking (BPR [26]) or the BCE loss.
- **LightGCN [Collaborative Filtering]** [10]: Building on NGCF [42], LightGCN is a neural graph-based approach that has fewer redundant neural components compared with the NGCF model, which makes it more efficient and effective.

To represent a user similarity-based approach, we implement a user encoding strategy inspired by the proposed hybrid approach of Taghavi et al. [39], albeit not relying on the use of agents, and using a different content-based model:

- **Customer Profile Similarity (CPS) [User Similarity]**: This strategy represents users in terms of a subset of the non-investment indicators concerning their retail transactions and balances with the bank. These indicators are weighted by age where possible, i.e. customer purchase features from two years ago have a lower weight than those from one year ago. Given these vector representations of each customer, we use these customer vectors to represent the financial assets. Specifically, for each financial asset, we assign it a vector representation that is the mean of the vectors of the customers that invested in that asset previously. The intuition here is that these aggregate vectors will encode the key indicators that make assets suitable for each customer. Finally, to score each asset for a new customer, we can simply calculate the cosine similarity between that customer and the aggregate asset representation, as both exist in the same dimensional space, where a higher similarity indicates the asset is more suitable for the customer.

Meanwhile, since it makes intuitive sense to use profitability as a surrogate for suitability in this context, we also include a range of key performance indicator predictors, which rank financial assets by their predicted profitability:

- **Predicted Profitability (PP) [KPI Predictor]**: This approach instead of attempting to model the customer–asset relationship instead attempts to predict an aspect of each asset alone, namely its profitability. In particular, three regression models are utilized to predict profitability of the assets over the aforementioned test windows. These models utilize features descriptive of the performance of the asset over incremental periods of time prior to testing, namely, [3,6,9] months prior, and include volatility, average closing price and expected returns over these periods.
  - **Linear Regression (LR)** [29]: Linear regression attempts to determine a linear relationship between one or more independent variables and a dependent variable by assigning these independent variables coefficients. It seeks to minimize the sum of squares between the observations in the dataset and the values predicted by the linear model.
  - **Support Vector Regression (SVR)** [34]: Support vector regression is a generalization of support vector machines for continuous values, wherein the aim is to identify the tube which best approximates the continuous-valued function. The tube is represented as a region around the function bounded on either side by support vectors around the identified hyperplane.
  - **Random Forest Regression (RFR)** [30]: Random forest regression is an ensemble method that utilizes a diverse array of decision tree estimators and conducts averaging upon their results in a process that significantly reduces the variance conferred by individual estimators by introducing randomness. Decision trees, in turn, are models that infer decision rules in order to predict a target variable.

In addition, we also implement a range of hybrid models that combine the aforementioned collaborative filtering approaches with customer and item intrinsic and historical features, in a similar manner to the experiments conducted by Taghavi et al. [39], Luef et al. [17] and others who use the intuition of combining intrinsic attributes of customers and items alongside collaborative filtering inputs:

- **MF+CI [Hybrid]**: This model combines the ranked recommendation lists of matrix factorization and the customer similarity model using rank aggregation. The specific rank aggregation method used is score voting.

- **LightGCN+CI [Hybrid]**: This model combines the ranked recommendation lists of LightGCN and the customer similarity model using rank aggregation. The specific rank aggregation method used is score voting.

Training Setup To train our supervised models, we need to divide the dataset into separate training and test subsets. We use a temporal splitting strategy here, where we define a time point to represent the current moment where investment recommendations are being requested. All data prior to that point can be used for training, while the following 9 months of data can be used to evaluate success. We create three test scenarios in this manner, where the three time points are 1st of October 2019 (denoted 2019), 1st of April 2020 (denoted 2020A) and the 1st of July 2020 (denoted 2020B). The collaborative filtering models are each trained for 200 epochs. Due to the large size of the dataset, a batch size of 100000 is chosen for training in order to speed up training time. Each model is run 5 times, and performance numbers are averaged out across the 5 runs.

Metrics To quantitatively evaluate how effective financial asset recommendation is, we want to measure how satisfied a customer would be if they followed the investment recommendations produced by each approach. Of course, this type of evaluation is not possible in practice, since we would need to put each system into production for an extended period of time (e.g. a year). Instead, we rely on surrogate metrics that are more practically available and capture different aspects of how satisfied a user might be if they followed the recommendations produced. There are two main aspects of our recommendations that we examine here:

- *Investment Prediction Capability*: First, we can consider how well the recommendations produced by an approach match what a customer actually invested in. The assumption here is that if the customers are making intelligent investments, then the recommendation system should also be recommending those same financial assets. Hence, given a customer and a point in time, where that customer has invested in one or more assets after that time point, we can evaluate whether the recommendation approach included those assets that the customer invested in within the top recommendations. We use the traditional ranking metric normalized discounted cumulative gain (NDCG) to measure this [12]. NDCG is a top-heavy metric, which means that an approach will receive a higher score the closer to the top of the recommendation list a relevant financial asset is. NDCG performance is averaged across all users in the test period considered. There are two notable limitations with this type of evaluation, however. First, it assumes that the customer is satisfied with what they actually invested in, which is not always the case. Second, only a relatively small number of users have investment history, so the user set that this can evaluate over is limited.
- *Investment Profitability*: An alternative approach for measuring the quality of the recommendations is to evaluate how much money the customer would have made if they followed the recommendations produced by each approach. In this case, for an asset, we can calculate the return on investment when investing in the top few recommended assets after a year, i.e. annualized return. The issue with this

type of metric is it ignores any personalized aspects regarding the customer's situation and can be highly volatile if there are assets who's price experiences significant growth during the test period (as we will show later).

For both of these aspects, we calculate the associated metrics when analysing the top '*k*' assets recommended, where '*k*' is either 1, 5 or 10.

6.2 Investment Recommendation Suitability

In this section, we will report and analyse the performance of the aforementioned financial asset recommendation models when applied to the Greek Stock market for customers of the National Bank of Greece. The goal of this section is not to provide an absolute view of asset recommendation approach performance, since there are many factors that can affect the quality of recommendations, most notably the characteristics of the market in question for the time period investigated. Instead, this analysis should be used as an illustration of how such systems can be evaluated and challenges that need to be considered when analysing evaluation output.

Investment Prediction Capability We will start by evaluating the extent to which the recommendation approaches recommend the assets that the banking customers actually invested in. Table 10.2 reports the investment prediction capability of each of the recommendation models under NDCG@[1,5,10] for each of the three scenarios (time periods tested). The higher NDCG values indicate that the model

Table 10.2 Financial asset recommendation model performances when evaluating for investment prediction capability. The best performing model per scenario (time period) is highlighted in bold

Scenario	Approach type	Recommender model	NDCG@1	NDCG@5	NDCG@10
2019	Collaborative filtering	MF	0.292	0.219	0.187
		LightGCN	**0.447**	**0.351**	**0.291**
	User similarity	CustomerSimilarity	0.026	0.009	0.007
	Hybrid	MF+CI	0.372	0.189	0.132
		LightGCN+CI	0.427	0.229	0.160
2020A	Collaborative filtering	MF	0.336	0.243	0.199
		LightGCN	**0.461**	**0.349**	**0.283**
	User similarity	CustomerSimilarity	0.011	0.012	0.015
	Hybrid	MF+CI	0.376	0.190	0.139
		LightGCN+CI	0.409	0.222	0.161
2020B	Collaborative filtering	MF	0.359	0.263	0.223
		LightGCN	**0.462**	**0.358**	**0.294**
	User similarity	CustomerSimilarity	0.012	0.012	0.011
	Hybrid	MF+CI	0.355	0.185	0.135
		LightGCN+CI	0.390	0.218	0.158

is returning more of the assets that the customers invested within in the top ranks. The highest performing model per scenario is highlighted in bold. Note that we are omitting the KPI prediction models from this analysis, as they can (and often do) recommend items that no user has invested in, meaning that their NDCG performance is not truly comparable to the other models under this type of evaluation.

As we can see from Table 10.2, of all the models tested, the type of model that produces the most similar assets to what the customers actually choose to invest in is the collaborative filtering type models, where the more advanced graph-based neural model (LightGCN) is better than the classical matrix factorization (MF) model. Given that collaborative filtering models aim to extract patterns regarding past investments across customers, it indicates the customers in our dataset largely invest in similar items over time (hence there are patterns that the collaborative filtering approaches are learning). On the other hand, the user similarity-based model recommends very different assets to what the customers invested within, indicating that the customers do not cluster easily based on the features we extracted about them. Finally, we observe that the hybrid models that integrate more features about the customers and items result less similar assets being recommended than the stock collaborative filtering only approaches. This likely means that the dataset is too small (i.e. has too few examples) to confidently extract generalizable patterns from the large number of new features being given to the model here.

However, just because the collaborative filtering models have the highest scores here does not necessarily mean that they will be the best. Recall that this metric is capturing whether the model is producing the same assets as what the customer actually invested in, not whether those were effective investments. Hence, we will next examine the profitability of the assets recommended.

Investment Profitability We next evaluate the profitability of the recommended assets in order to assess the effectiveness of the recommendation models. Profitability is evaluated as the expected return over each 9-month testing period and then annualized. Table 10.3 reports annualized profitability when considering the top [1,5,10] recommended financial assets by each recommendation model. Profitability is calculated as a percentage return on investment over the year period.

From Table 10.3, we can observe some interesting patterns of behaviour between the different types of recommendation models. Starting with the collaborative filtering-based approaches, we can see that for the 2019 test scenario, these models would have lost the customer a significant amount of value. However, for the 2020A/B scenarios, these models would have returned a good profit of around 30%. This indicates that there was a marked shift in the types of financial asset that was profitable between 2018 and 2020. It also highlights a weakness of these collaborative filtering models, in that they assume that past investments were profitable and hence similar investments now will also be profitable, which was clearly not the case between 2018 (initial training) and the 2019 test scenario. Furthermore, we also see the hybrid models that are based on the collaborative filtering strategies suffering from the same issue, although it is notable that when

Table 10.3 Financial asset recommendation model performances when evaluating for asset profitability. The best performing model per scenario (time period) is highlighted in bold

Scenario	Approach type	Recommender model	Return@1	Return@5	Return@10
2019	Collaborative filtering	MF	−28.76%	−39.51%	−30.71%
		LightGCN	−29.4%	−37.99%	−25.42%
	User similarity	CustomerSimilarity	3.7%	−0.84%	−0.5%
	Hybrid	MF-CI	−42.72%	−42.43%	−42.43%
		LightGCN-CI	−43.65%	−43.07%	−43.07%
	KPI prediction	LR	**114.05%**	**40.37%**	15.42%
		SVR	14.6%	36.61%	29.19%
		RFR	26.23%	39.02%	**31.04%**
2020A	Collaborative filtering	MF	35.7%	34.66%	34.44%
		LightGCN	**36.74%**	22.99%	20.65%
	User similarity	CustomerSimilarity	21.69%	23.59%	23.94%
	Hybrid	MF-CI	43.31%	42.89%	42.89%
		LightGCN-CI	44.91%	39.61%	39.6%
	KPI prediction	LR	0.52%	10.49%	59.98%
		SVR	22.03%	17.02%	**64.05%**
		RFR	0.52%	**59.46%**	31.45%
2020B	Collaborative filtering	MF	56.08%	52.29%	50.34%
		LightGCN	**76.72%**	43.44%	35%
	User similarity	CustomerSimilarity	21.89%	23.83%	23.53%
	Hybrid	MF-CI	75.94%	77.11%	77.11%
		LightGCN-CI	82.82%	74.16%	74.11%
	KPI prediction	LR	−20.15%	**291.04%**	**176.97%**
		SVR	−1.31%	−1.31%	39.21%
		RFR	−3.92%	−6.33%	36.8%

they are working as intended, they recommend more profitable assets than the models that use collaborative filtering alone, showing that the extra features that were added are helping identify profitable asset groups. In terms of the user similarity-based model, we again see poorer performance in the 2019 scenario than the 2020A/B scenarios, which is expected as this type of model is based on analysis of the customers investment history. However, the returns provided by this model are reasonably consistent (around 20–23%) in 2020.

Next, if we compare to the KPI Predictor models that are attempting to explicitly predict assets that will remain profitable, we see that these models can be quite volatile. For instance, for the 2019 test scenario and the LR model, the top recommended asset exhibited a return of 114%, but that same model in 2020B placed an asset in the top rank that lost 20% of its value. This volatility is caused since these models are relying on (recent) past performance of an asset remaining a strong indicator of its future performance. For example, if we consider the 2020B scenario with the 20% loss for the top-ranked asset, that is a case where the asset's price had been recently inflated, but dropped rapidly during the test period. However, if we consider the KPI predictors holistically, as a general method for building a broader portfolio (e.g. if we consider the returns for the top 10 assets), these approaches seem to be consistently recommending highly profitable assets in aggregate and generally recommend more profitable assets than the collaborative filtering-based approaches.

Overall, from these results, we can see some of the challenges when evaluating models automatically without a production deployment. Measuring investment prediction capacity of a model can provide a measure of how similar the recommendations produced are to how the customers choose to invest. However, our analysis of profitability indicates that there is no guarantee that these are good investments from the perspective of making a profit. On the other hand, we can clearly see the value that these automatic recommender systems can bring for constructing portfolios of assets, with some quite impressive returns on investment when averaged over the top-10 recommended assets.

7 Summary and Recommendations

In this chapter, we have provided an introduction and overview for practitioners interested in developing financial asset recommendation systems. In particular, we have summarized how an asset recommendation system functions in terms of its inputs and outputs, discussed how to prepare and curate the data used by such systems, provided an overview of the different types of financial asset recommendation model, as well as their advantages and disadvantages, and finally included an analysis of those models using a recent dataset of assets from the Greek stock market and customers from a large bank. Indeed, we have shown that this type of recommendation system can provide effective lists of assets that are highly profitable, through the analysis of historical transactional and asset pricing data.

For new researchers and practitioners working in this area, we provide some recommendations below for designing such systems based on our experience:

- **Data cleanliness is critical**: Good data preparation allows for efficient analysis, limits errors, eliminates biases and inaccuracies that can occur to data during processing and makes all of the processed data more accessible to users. Without sufficient data preparation, many of the models discussed above will fail due to noisy data. For instance, we needed to spend significant time removing pricing outliers from the asset pricing dataset due to issues with the data provider.
- **Understand what data you can leverage**: There are a wide range of different types of information that you might want to integrate into an asset recommendation model: past investments, asset pricing data, customer profiles and market data, among others. However, often some of this data will be unavailable or too sparse to be usable. Pick a model that is suitable for the data you have available, and avoid trying to train models with too few examples.

- **Analyse your models thoroughly**: Sometimes trained machine learned models hallucinate patterns that are not there, simply because they have not seen counter examples to sufficiently calibrate, particularly in scenarios like this where assets can exhibit short price spikes or troughs. For instance, we have seen one of our KPI predictors suggests that an asset could have an annualized return in the thousands of %, which is obviously impossible. You may wish to apply additional filtering rules to your machine learned models to remove cases where the model is obviously miss-evaluating an asset and use multiple metrics like those presented here to quantify how well those models are functioning before putting them into production.

Acknowledgments The research leading to these results has received funding from the European Community's Horizon 2020 research and innovation programme under grant agreement no: 856632 (INFINITECH).

References

1. Aggarwal, C. C., et al. (2016). *Recommender systems* (Vol. 1). Springer.
2. Agrawal, R., Imieliński, T., & Swami, A. (1993, June). Mining association rules between sets of items in large databases. *SIGMOD Record, 22*(2), 207–216.
3. Bennett, J., Lanning, S., et al. (2007). The Netflix prize. In *Proceedings of KDD cup and workshop* (Vol. 2007, p. 35). Citeseer.
4. Cragin, M. H., Heidorn, P. B., Palmer, C. L., & Smith, L. C. (2007). An educational program on data curation.
5. Curry, E., Freitas, A., & O'Riáin, S. (2010). The role of community-driven data curation for enterprises. In *Linking enterprise data* (pp. 25–47). Springer.
6. Felfernig, A., Isak, K., & Russ, C. (2006). Knowledge-based recommendation: Technologies and experiences from projects. In *Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva Del Garda, Italy*, NLD (pp. 632–636). IOS Press.
7. Freitas, A., & Curry, E. (2016). Big data curation. In *New horizons for a data-driven economy* (pp. 87–118). Cham: Springer.
8. Ginevičius, T., Kaklauskas, A., Kazokaitis, P., & Alchimovienė, J. (2011, September). Recommender system for real estate management. *Verslas: teorija ir praktika, 12*, 258–267.
9. Gonzalez-Carrasco, I., Colomo-Palacios, R., Lopez-Cuadrado, J. L., Garciá-Crespo, Á., & Ruiz-Mezcua, B. (2012). Pb-advisor: A private banking multi-investment portfolio advisor. *Information Sciences, 206*, 63–82.
10. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proc. of SIGIR* (pp. 639–648).
11. Ho, G., Ip, W., Wu, C.-H., & Tse, M. (2012, August). Using a fuzzy association rule mining approach to identify the financial data association. *Expert Systems with Applications, 39*, 9054–9063.
12. Järvelin, K., & Kekäläinen, J. (2017). IR evaluation methods for retrieving highly relevant documents. In *ACM SIGIR Forum* (Vol. 51, pp. 243–250). New York, NY: ACM.
13. Kay, J. (2012). The Kay review of UK equity markets and long-term decision making. *Final Report, 9*, 112.
14. Khusro, S., Ali, Z., & Ullah, I. (2016). Recommender systems: Issues, challenges, and research opportunities. In K. J. Kim & N. Joukov (Eds.), *Information Science and Applications (ICISA)*

2016, Singapore (pp. 1179–1189). Singapore: Springer.

15. Kumar, P., & Thakur, R. S. (2018). Recommendation system techniques and related issues: a survey. *International Journal of Information Technology, 10*(4), 495–501.
16. Lee, E. L., Lou, J.-K., Chen, W.-M., Chen, Y.-C., Lin, S.-D., Chiang, Y.-S., & Chen, K.-T. (2014). Fairness-aware loan recommendation for microfinance services. In *Proceedings of the 2014 International Conference on Social Computing (SocialCom '14)* (pp. 1–4). New York, NY: Association for Computing Machinery.
17. Luef, J., Ohrfandl, C., Sacharidis, D., & Werthner, H. (2020). A recommender system for investing in early-stage enterprises. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing, SAC '20* (pp. 1453–1460). New York, NY: Association for Computing Machinery.
18. Ma, H., Yang, H., Lyu, M. R., & King, I. (2008). SoRec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (pp. 931–940).
19. Manotumruksa, J., Macdonald, C., & Ounis, I. (2017). A deep recurrent collaborative filtering framework for venue recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 1429–1438).
20. Matsatsinis, N. F., & Manarolis, E. A. (2009). New hybrid recommender approaches: An application to equity funds selection. In F. Rossi & A. Tsoukias (Eds.), *Algorithmic Decision Theory* (pp. 156–167). Berlin: Springer.
21. Mitra, S., Chaudhari, N., & Patwardhan, B. (2014). Leveraging hybrid recommendation system in insurance domain. *International Journal of Engineering and Computer Science, 3*(10), 8988–8992.
22. Musto, C., Semeraro, G., Lops, P., de Gemmis, M., & Lekkas, G. (2014, January). Financial product recommendation through case-based reasoning and diversification techniques. *CEUR Workshop Proceedings* (Vol. 1247).
23. Nair, B. B., Mohandas, V. P., Nayanar, N., Teja, E. S. R., Vigneshwari, S., & Teja, K. V. N. S. (2015). A stock trading recommender system based on temporal association rule mining. *SAGE Open 5*(2). https://doi.org/10.1177/2158244015579941
24. Niu, J., Wang, L., Liu, X., & Yu, S. (2016). FUIR: Fusing user and item information to deal with data sparsity by using side information in recommendation systems. *Journal of Network and Computer Applications, 70*, 41–50.
25. Paranjape-Voditel, P., & Deshpande, U. (2013). A stock market portfolio recommender system based on association rule mining. *Applied Soft Computing, 13*(2), 1055–1063.
26. Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In *Proc. of UAI* (pp. 452–461).
27. Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2010). Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web* (pp. 811–820).
28. Rendle, S., Krichene, W., Zhang, L., & Anderson, J. (2020). Neural collaborative filtering vs. matrix factorization revisited. In *Proc. of RecSys* (pp. 240–248).
29. Schneider, A., Hommel, G., & Blettner, M. (2010, November). Linear regression analysis part 14 of a series on evaluation of scientific publications. *Deutsches Ärzteblatt International, 107*, 776–782.
30. Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal, 20*(1), 3–29.
31. Schumaker, R. P., & Chen, H. (2009, March). Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems, 27*(2), 1–19.
32. Sehgal, V., & Song, C. (2007). Sops: Stock prediction using web sentiment. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)* (pp. 21–26).
33. Seo, Y.-W., Giampapa, J., & Sycara, K. (2004). Financial news analysis for intelligent portfolio management.
34. Shmilovici, A. (2005). *Support vector machines* (pp. 257–276). Boston, MA: Springer US.

35. Singh, I., & Kaur, N. (2017). Wealth management through Robo advisory. *International Journal of Research-Granthaalayah, 5*(6), 33–43 (2017)
36. Smith, B., & Linden, G. (2017). Two decades of recommender systems at amazon.com. *IEEE Internet Computing, 21*(3), 12–18.
37. Steck, H. (2011). Item popularity and recommendation accuracy. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (pp. 125–132).
38. Swezey, R. M. E., & Charron, B. (2018, September). Large-scale recommendation for portfolio optimization. In *Proceedings of the 12th ACM Conference on Recommender Systems*.
39. Taghavi, M., Bakhtiyari, K., & Scavino, E. Agent-based computational investing recommender system. In *Proceedings of the 7th ACM Conference on Recommender Systems* (pp. 455–458).
40. Tian, G., Wang, J., He, K., Sun, C., & Tian, Y. (2017). Integrating implicit feedbacks for time-aware web service recommendations. *Information Systems Frontiers, 19*(1), 75–89.
41. Vasile, F., Smirnova, E., & Conneau, A. (2016). Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems* (pp. 225–232).
42. Wang, X., He, X., Wang, M., Feng, F., & Chua, T.-S. (2019). Neural graph collaborative filtering. In *Proc. of SIGIR* (pp. 165–174).
43. Wu, C., & Yan, M. (2017). Session-aware information embedding for e-commerce product recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 2379–2382).
44. Yujun, Y., Jianping, L., & Yimei, Y. (2016, January). An efficient stock recommendation model based on big order net inflow. *Mathematical Problems in Engineering, 2016*, 1–15.
45. Zhao, X., Zhang, W., & Wang, J. (2015). Risk-hedged venture capital investment recommendation. In *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15* (pp. 75–82). New York, NY: Association for Computing Machinery.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.