MFS Investment Management

MIT Business Analytics Capstone

Milestone 5: Final Report

Iris Brook, Xander Pero

MFS Advisors: Brain Shaw, Suzet Nkwaya, Erin Haley

MIT Advisor: James Butler

Contents

Company and Team Overview

MFS Investment Management, also known as Massachusetts Financial Services, stands as a leading global asset manager. Established in 1924 and headquartered in Boston, MA, MFS distinguishes itself as one of the world's most respected asset management entities, boasting a legacy of innovation, notably pioneering the mutual fund which has left a significant mark on the investment landscape. As of February 29, 2024, the firm oversees $615.1 billion in assets, reflecting its significant impact and trust within the financial community. MFS employs over 350 sales professionals, demonstrating a strong commitment to client service. There are both internal sales professionals that primarily work in-office, and external sales professionals that primarily set up in-person meetings. MFS offers a wide array of investment products, including equity, fixed income, and multi-asset strategies, catering to a diverse client base from around the world through offices in London, Tokyo, and Singapore. At the heart of MFS's success is a research-driven investment approach through their partnership with MIT, emphasizing thorough analysis to shape its strategic decisions, therefore ensuring sustained growth and client satisfaction in a dynamic global market.

Our project resides within MFS Investment Management's Global Distribution team, specifically under the Business Intelligence division. Operating within the Global Distribution Strategy Team, led by Nadine Kawkabani, we report to Brian Shaw, and collaborate with Suzet Nkwaya and Erin Haley. This team structure positions our project at the intersection of data-driven insights and strategic decision-making within the asset management sector, focusing on enhancing client interactions and optimizing sales strategies through advanced analytics and tailored recommendations.

Project Context

At MFS, the sales team plays a pivotal role in driving client engagement and relationship management strategies. Clients are financial advisors who manage their own book of people's investments. Tasked with fostering strong partnerships with financial advisors globally, the sales team acts as the primary liaison between MFS and its network of financial advisors. These advisors rely on MFS for their investment solutions.

MFS empowers its sales team with extensive resources and support to effectively communicate the firm's investment philosophies and product offerings. Primarily, sales team employees utilize dashboards within Salesforce Customer 360 (C360) to brief themselves on previous interactions, current holdings, and additional information of financial advisors when preparing for meetings. MFS employees tailor interaction topics to the diverse needs of their client.

The sales team's proactive engagement extends beyond initial client onboarding to include ongoing relationship maintenance and strategic consultation. By staying on top of market trends and leveraging MFS's research-driven insights, the team ensures that advisors have access to the latest information and resources needed to make informed investment decisions. This collaborative partnership model not only strengthens client trust and loyalty but also reinforces MFS's reputation as a trusted steward of client assets.

Problem Statement and Opportunity

We identified the problem statement as the need to enhance efficiency within the MFS sales team by addressing several critical issues: time-consuming review of previous interactions, disorganized data across multiple tabs, and reliance on employees' judgment for topic selection. These challenges collectively hinder effective communication with financial advisors.

The primary issue arises from sales team employees having to individually examine all previous interactions to understand an advisor's communication history. This process is further complicated by dispersed and disorganized data, significantly increasing the time and workload required to access and digest necessary information. Addressing these challenges is crucial for improving time efficiency, ensuring consistency in client engagements and enabling data-driven decision making. By reducing time spent on review, we can increase productive client-facing time. A standardized approach to summarizing interactions and recommending topics can lead to more consistent, high-quality engagements across the entire sales team.

We aim to develop a tool for the MFS sales team that for a given financial advisor, can summarize previous interactions and offer tailored recommendations on discussion points to maximize successful interactions and therefore improve client relationships.

This project presents several key opportunities: maximizing successful interactions, improving client relations, and ultimately increasing sales. By leveraging historical interactions and sales data more efficiently, sales team members can demonstrate a deeper understanding of each advisor's needs and preferences. We anticipate measurable improvements in key metrics, such as reduction in preparation time, increase in successful interactions, and long-term growth in assets under management or new account acquisitions.

Once implemented, MFS can use its results to continuously improve recommendations. With this project, we are both solving immediate challenges and developing the path for long-term improvements in how MFS interacts with its financial advisors, moving from intuition-based to data-driven strategies.

Data Sources

MFS has access to each financial advisor's contact information, investment history, previous interactions with sales team members. We were provided with extracts of these datasets at the start of our project with updates as needed. The personal information highlights work-related attributes, such as firm, tenure, and location.

Two tables relate to investment history: flows and assets under management (AUM). The flows indicate each purchase, redemption, transfer, or exchange order within MFS funds, dating back to 2015. This highlights the specific fund, date, and financial advisor client executing the order. Purchases minus redemptions is one metric to calculate the change in investment while ignoring transfers or exchanges between funds or advisors. On the other hand, AUM is a monthly snapshot of an advisor's total holdings split across funds. These values increase or decrease along with the fund's performance. AUM is the best representation of total investment across MFS funds.

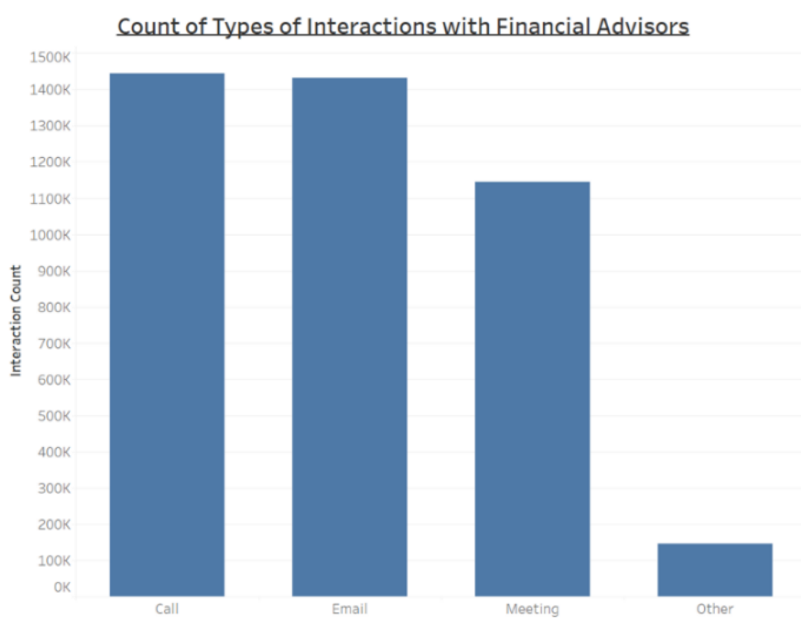
Interactions are characterized by a unique ID. They describe the method and date of communication, list of attendees, and meeting notes from the interaction. These notes are either dictated or typed by the sales team employee. Dictated notes are transcribed by a third-party company called Hey Dan. Some data challenges we encountered were abbreviations, misspellings, words with dual meanings, empty notes, and non-descriptive notes.

Other pieces of data we chose not to include were MFS fund performance data and overall market trends. These could be useful to remove positive or negative trends within a person's holdings. However, we chose to use purchases minus redemptions, which are less affected by these variables than an advisor's AUM. Moreover, performance and market data would be used to de-trend the flow data and is not directly required as an input to the recommendation system. Such approaches were deemed less important than development of the summarization model and recommendation system.

Exploratory Data Analysis

To develop a comprehensive understanding of the interactions between MFS sales team members and financial advisors, we conducted an in-depth exploratory data analysis (EDA). This process involved examining various aspects of the interaction data to identify key patterns and insights. The EDA is crucial as it provided the foundation for developing our effective summarization and recommendation tool tailored to the specific needs and behaviors of the sales team and financial advisors. Below are some of the key findings from our EDA, illustrated through three important charts.

1. Interaction Types with Financial Advisors:

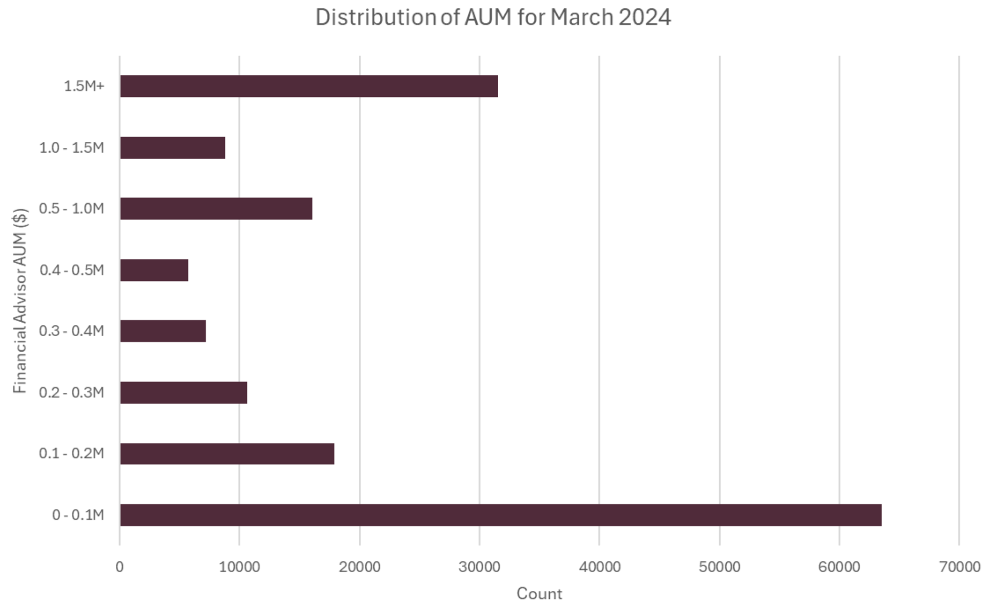


This chart illustrates the distribution of different types of interactions between MFS sales team members and financial advisors, including calls, emails, meetings, and other forms of communication. The data indicates that most interactions are conducted through calls and emails, which are less costly and time-consuming compared to in-person meetings. This insight is crucial as it highlights the predominant modes of communication and can inform the development of tools and resources to enhance efficiency in these areas. By understanding the frequency and nature of these interactions, we can better tailor our summarization and recommendation tools to the most common communication channels, ensuring they provide maximum utility to the sales team.

2. Most Frequent Topics in Interactions:

This chart presents the most frequently discussed topics during interactions with financial advisors. Topics such as Fixed Income, Business Building, and Competitive Opportunity Review are among the most common. This information is vital for the development of our recommendation model as it helps identify which topics are frequently relevant and should be prioritized. By focusing on these prevalent topics, the recommendation algorithm can suggest discussion points that are more likely to resonate with financial advisors and align with their interests and needs. Also, understanding the common themes can help train sales team members on key discussion areas and enhance their engagements' effectiveness.

3. Distribution of AUM (Assets Under Management) for March 2024:



The distribution chart of Assets Under Management (AUM) for March 2024 (we chose a recent month to showcase current status) shows the spread of financial advisors based on the amount of assets they manage. The majority of financial advisors manage portfolios with lower AUM, while a smaller number manage larger portfolios. This data is essential for segmenting financial advisors and tailoring interaction strategies accordingly. Additionally, by understanding the AUM distribution, the sales team can prioritize and customize their engagement efforts based on the value and potential impact of each advisor. For instance, high-AUM advisors may require more personalized and high-touch interactions, whereas lower-AUM advisors might benefit from more standardized support and resources.

Solution Methodology

Initial Scope and Methodology

Our scope of work involves two main points. First, we will summarize relevant data (personal, interaction, etc.) by identifying previously discussed topics and highlights. We will work alongside MFS sales team members and historical data to determine an exhaustive list of potential topics. Second, we will recommend future discussion topics according to our recommendation algorithm. We will build a recommendation algorithm that recommends topics for each financial advisor using their interaction and investment history.

Additional Considerations

Ensuring data privacy is paramount, requiring strict adherence to data protection regulations to safeguard sensitive information. However, the quality of the data presents challenges, as many interaction notes are hand-filled or transcribed by a third-party service, leading to inconsistencies in spelling or content. This variation necessitates considerable time dedicated to cleaning the language portion of the data and negatively impacted our model's performance.

The adoption of the new system by the sales team is crucial for the project's success. Achieving the team's buy-in and ensuring they effectively use the system is essential for realizing the tool's potential benefits. Their feedback is useful for continuous improvement of the recommendation system, particularly from a usability standpoint, considering they are the end-users of it.

Short-term evaluation of the system's impact presents challenges. Quantifying success for our model is complex, given that engagement or increased investment in MFS portfolios is a long-term objective and may be difficult to directly attribute to specific initiatives. Due to the indirect relationship between the project's outcomes and investment inflow, it could take time to observe the benefits of our efforts.

Interaction Meeting Note Cleaning

To mitigate the challenge of topic misclassification due to data inconsistencies, several preprocessing steps were implemented. We removed HTML and Unicode text artifacts from the data to ensure uniformity and clarity of content. Additionally, we replaced apostrophes, commas, and other non-semantic characters with whitespace. A critical enhancement involved identifying and marking abbreviations and acronyms within the text corpus. This step was crucial for aiding the model in accurately identifying keywords, and not components of longer words, reducing the number of keywords found erroneously.

Solution Overview

Our solution consists of two primary components designed to optimize client interactions between MFS sales team members and financial advisors. The first component, the summarization model, searches for keywords within meeting notes to classify the corresponding topics in meetings between sales members and financial advisors. Once each interaction is associated with zero or more topics, we can analyze and attribute its success to the underlying topics. Success can be defined by an increase in relationship strength, or by proxy, an increase in investment in MFS portfolios. The recommendation model generates actionable advice for a chosen financial advisor, suggesting a list of potential discussion topics for the sales team member to consider in future interactions. This holistic approach aims to enhance the relevance and effectiveness of client meetings, thereby fostering stronger client relationships and business growth opportunities for MFS.

A secondary component is retrieving personal preferences from the meeting notes. Using a separate lexicon, we search for non-business information, such as hobbies, preferences, and icebreaker information. This component is not as important as the sales topic recommendation but serves to build out a more complete profile for each financial advisor. These efforts are highlighted in the non-topic meeting note summarization in the next section.

Topic Assignment Modelling

Keywords and Topic Mapping

The topic assignment model operates on a curated list of topics derived from extensive historical meeting data between MFS sales employees and financial advisors. We were given an original list of 477 keywords associated with 80 topics. We have added 20 additional keywords and one topic. Each keyword is mapped to a specific topic, such as "401k" for retirement insights or "climate change" for sustainable investing. Additionally, there is a type, which corresponds to multiple topics, and a broad category, which corresponds to multiple types. We currently do not use these in the modelling step, but they may be useful for grouping similar topics.

After preprocessing the meeting notes, the model searches the meeting notes for each keyword, labeling it with the associated topic if present. Topics are not duplicated even if two different keywords associated with it are present. If the interaction remains untagged after searching the meeting notes, it scans the subject line. Some employees write brief information in the subject line, especially if they leave the meeting note empty, for follow-up emails, missed calls, or small interactions.

Some interaction notes may remain untagged if no pertinent topics are identified, while others may be associated with multiple topics. This is due to the wide range of quality in each meeting note, from empty to overly descriptive with miscellaneous information.

Non-Topic Meeting Note Summarization

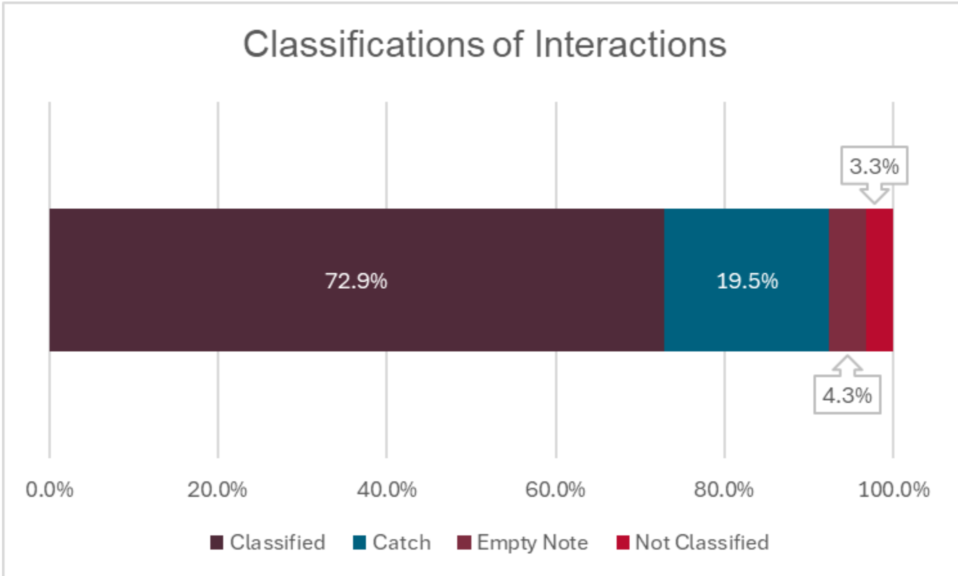
Our topic assignment use case involves accurately identifying underlying topics of each interaction. However, an additional enriching facet is summarizing all other pieces of information. Some examples are meeting preferences (dinners versus sports games), hobbies, and background information. Though not explicitly tied to how the client resonates with specific topics, this information could be useful in strengthening the relationship and establishing trust.

We began work on summarizing non-topic information, specifically in building a keyword dictionary with hobbies, meeting preferences, and background information. This showed promise initially but was limited in the scope of the keyword dictionary. Moreover, college names and some hobbies are common words, anecdotally resulting in more false positives than desired.

We stopped working on non-topic summarization because MFS had begun a project utilizing OpenAI's API to summarize meeting notes. This approach would render our work obsolete, so we decided to instead focus on other parts of our project that were independent of that work. For that reason, we do not report the results of this endeavor.

Investment Topic Assignment Model Performance

The chart below illustrates the classification results of our topic assignment model:



We successfully classified over 70% of the interaction notes, as indicated by the "classified" category, which demonstrates the efficacy of our model in accurately tagging relevant topics. The "catch" category, accounting for 19.5%, includes interactions related to scheduling meetings, voicemails, missed calls, out-of-office responses, and other similar content that lacked substantive discussion points. "Empty note" represents 4.3% of the notes that were entirely empty. The remaining 3.3% fell under "not classified," encompassing notes that the model could not categorize due to insufficient or ambiguous information. It's important to note that 16.6% of the meeting notes in the system are empty, but 12.3% were classifiable using the subject line, either into "classified" or "catch."

Limitations

We chose to stay with the keyword search, and not a machine learning approach, due to a few limitations. First, we don't have a ground truth labeled dataset depicting the true topics that were covered. To create this, we'd use the keyword search as a baseline, but this defeats the point. If we wanted to use advanced methods to fix spelling errors (rare within the dataset), this would still fall under preprocessing and not keyword search. Second, after examining the unclassified meeting notes, it's due to non-substantive notes rather than holes within the keyword search. Simply put, the keyword search algorithm fulfills the need for topic assignment.

Recommendation Modelling

A unique characteristic of the data we want to summarize and recommend is that they are two-way interactions. Most recommendation system problems are built from user-only data, such as their click history or search preferences. However, this is not the case with client interactions, especially under the assumption that the discussed topics are initiated by the sales team employee rather than the financial advisor. Therefore, we split our recommendation system into two steps.

First, we developed a success metric that reflects the impact of an interaction from the client's perspective. This was to offset the positive feedback loop that would occur if we built the recommendation system purely from the discussed topics, because it would reinforce the current behaviors of the sales team. We also evaluated results using the topics discussed, which can be interpreted as building recommendations to mirror the current logic of the sales team employees.

Second, we employed a recommendation algorithm. We chose an embarrassingly shallow autoencoder (EASE) referenced in Appendix 1. EASE can be interpreted as a collaborative-filtering method, modeled as a neural network with zero hidden layers, simplifying to a neighborhood-based approach linear model with closed-form. We implemented the recommendation system with the original topics and success metric separately, comparing results.

Client Filtering

We considered post-COVID data, particularly from March 2020 and onwards. We subset financial advisors with five or more interactions since then, as well as at least three topics. We created this split to train and evaluate on a better subset: actively engaged clients that MFS currently prioritizes. Advisors with few interactions or topics would not contribute useful information towards recommendation a topic versus another.

Metric A: Original Topic Counts

The naive metric simply uses the binary indicator if a topic was discussed. We chose to binarize the number of topics rather than use the number of times discussed. We believed that this would be easier to compare with post-interaction investment and create a robust model. Moreover, some employees send follow-up emails the next day, which double tags the specific topic, potentially throwing off numbers.

The main drawback of this metric is misalignment with the client's perspective. Because most interactions are planned by the sales team, it's likely that the associated topics are as well. Therefore, training the recommendation system with this metric would incline it to recommend topics according to the historical beliefs of the sales team. While this is useful, it would fail to discern good and bad topics from the client's perspective.

We create the client–day–topic tensor, initializing at zero and replacing it with one if, for each client, the topic was discussed on that day. To create the client–topic format for the recommendation system, we sum over all dates. Finally, we clip all positive values equal to one, ensuring a binary matrix.

Metric B: Post-Interaction Investment

The success metric we developed utilizes an advisor's change in investment after an interaction. We assume that investment increase and topic effectiveness are directly related. This is plausible but does not directly represent the underlying behavior of the system. Leadership at MFS believes that interactions are intended to build a trusting relationship with a client rather than act as a sales pitch. Then, because clients trust MFS, they will invest in its portfolios. Therefore, interactions do not cause immediate sales lift in the same way that marketing efforts in other industries could.

Overlooking the relationship aspect is our best proxy for determining the influence of an interaction without sentiment from the client's perspective. Though this isn't entirely true to the system, it aligns with how MFS makes money. We did not attempt sentiment analysis within the interaction's meeting notes due to their varying quality. MFS has attempted to apply sentiment analysis on meeting notes in the past with little success.

Approach 1: Transformer

We have two implementations that represent the monetary increase after an interaction. The first uses a time series transformer (Appendix 2) to forecast a client's investment using historical investment and interaction information. We tried using total AUM and purchases minus redemptions to represent the monetary information. Though this trains on historical nature, it is difficult to understand how each topic independently affects the change in forecasted AUM. We also obtained semi-poor results in forecasting fit using the transformer, reducing confidence in the ability to estimate specific topic effects independently. We decided to scrap this approach, instead focusing on a hand-crafted metric rather than a machine learning model. Nonetheless, the transformer implementation is coded for further development if desired.

Approach 2: Exponentially Decaying Flow Attribution

The second method is a non-machine learning approach. Using purchases minus redemptions as flow, it attributes flow to recent topics while adhering to some desired properties:

1. The flow attributed to one or more topics cannot exceed the raw value of the flow on a given day. For example, if an advisor has a net purchase of $500, at most $500 can be attributed to any recent topics.
2. A decreasing amount of flow should be attributed to topics over time. For example, if there is a net purchase of $500 on February 1 and another $500 on February 2, the attributed value on February 2 should be less than that on February 1.
3. Recent interactions do not completely overwrite the influence of previous interactions but should have a higher weighting than previous interactions.

We maintained these two properties by constructing an exponentially decaying window, beginning at 1 and ending at 0.01, over 180 days. We also created a client–day–topic tensor, with dimensions of roughly 50,000 by 1,500 by 73, initially valued at zero. The pseudocode below describes the algorithm.

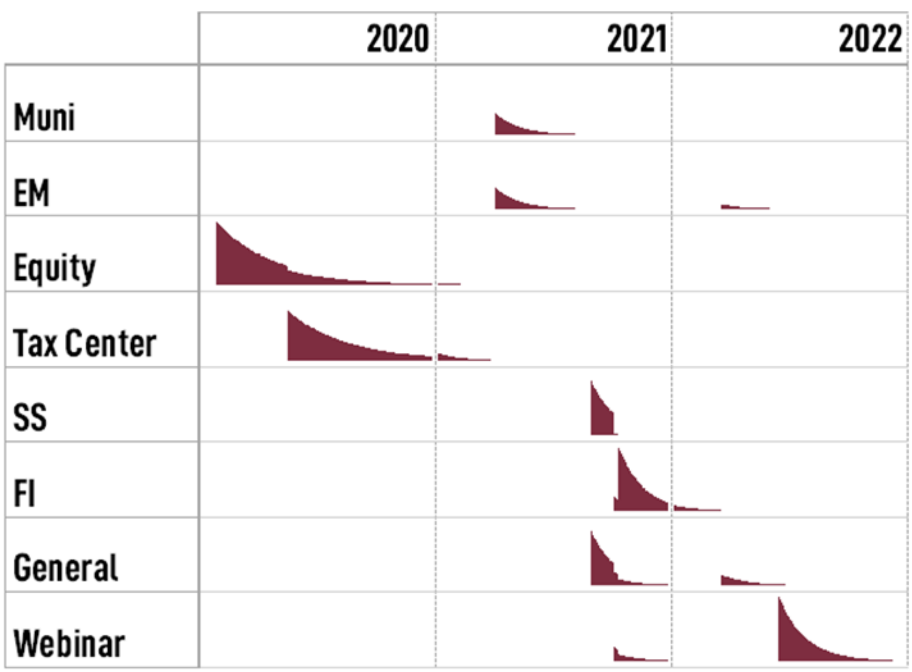
For each date:

- Find all clients with an interaction on that date, and the associated topics
- For each client–topic slice, forward fill 180 days into the future equal to the exponentially decaying window
- Calculate the total attribution over all topics individually per client on current day
- Divide future forward-filled values by the total attribution

The algorithm creates a curve that, when summed over all possible topics, does not exceed 1. This ensures property one is true. Exponential decay maintains property two.

Property three is maintained during the division by total attribution step. The total attribution includes topics from previous interactions, meaning that the total attribution is usually not equal to the number of new topics on the current day. For example, if there is a total attribution of 0.6 in the system, then an interaction with a single topic occurs, the historical topics will retain an attribution of 0.375. The new topic will receive an attribution of 0.625 (each is divided by 1.6). The exponential decay is then maintained with these initial values, obeying property two.

The chart shows an example attribution curve split by topic for a single client. Sample topics are vertically displayed, with the timeline from 2020 to 2022 on the horizontal axis. Evidently, any number of topics (including zero) can be relevant on a given date. When multiple topics are present in the same interaction, their starting value is normalized such that total attribution does not exceed one. As desired, topic relevance is highest immediately after their introduction and decays over time, eventually reaching zero.



This algorithm can be interpreted as modeling the memory of a client. The memory is bounded between zero and one, with a client having greater memory of recent interactions than distant interactions. No matter how many topics are brought up, the percentage in memory cannot exceed 100%, and is split amongst the many topics. When flow is not attributed to a topic, conceptually, it can be attributed to MFS's reputation rather than a specific topic.

Once we create the topic attribution curve for each client, we elementwise multiply the sequence against the client's purchases minus redemptions. The output is a client–day–topic tensor of attributed flow values. To put this into the client–topic format as input to the recommendation system, we sum over all dates. Finally, we divide by the maximum absolute value per person, constricting all attribution values between negative one and one. This ensures equal relevance amongst clients with different flow magnitudes since their total AUM could be thousands or millions.

Recommendation Algorithm: EASE

We selected EASE as our recommendation algorithm for a few reasons. First. We do not have access to a Gurobi license, so we did not consider heavyweight optimization approaches. Second, EASE requires less computation and training time than other collaborative-filtering approaches. This allowed us to evaluate EASE rigorously with our metric permutations, namely with metric A/B, with/without clusters, and small variations. Third, EASE only contains a single hyperparameter, requiring significantly less effort in hyperparameter tuning. Finally, EASE achieved similar, if not better, results than other approaches, indicating its competitive viability.

The attached paper in the appendix introduces the original optimization problem, further highlights its rationale as a neighborhood-based approach, and proves its closed-form solution. We highlight the simple Python implementation of the closed-form solution of EASE below, initially presented in the paper.

Algorithm 1: Training in Python 2 using numpy

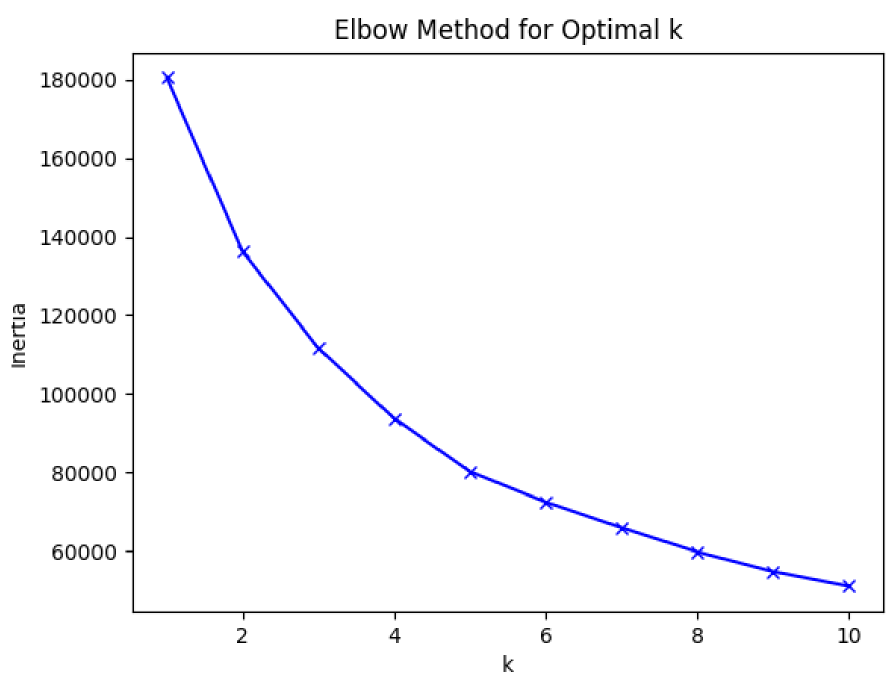
Input: data Gram-matrix $G := X^\top X \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$,
L2-norm regularization-parameter $\lambda \in \mathbb{R}^+$.
Output: weight-matrix B with zero diagonal
$diagIndices$ = numpy.diag_indices(G.shape[0])
$G[diagIndices]$ += λ
P = numpy.linalg.inv(G)
B = P / (-numpy.diag(P))
$B[diagIndices]$ = 0

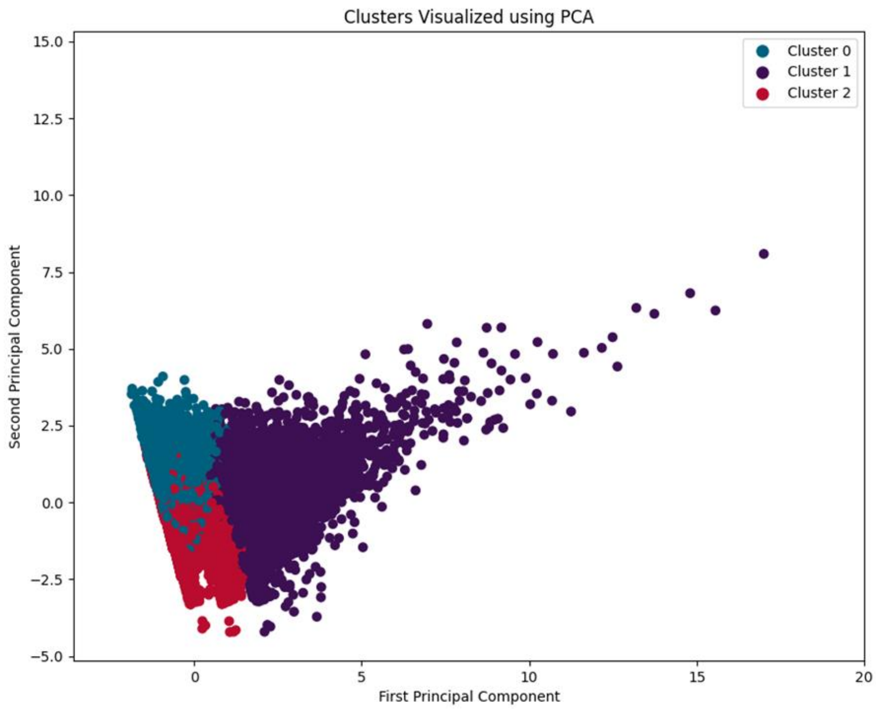
Clustering Clients

To enhance our recommendation system, we aimed to segment financial advisors into distinct groups, allowing for more tailored topic suggestions. By clustering advisors, we can provide recommendations that cater to the specific needs and characteristics of each group, rather than applying a one-size-fits-all approach across our entire dataset. We also wanted to incorporate additional demographic and client information that wasn't factored into the recommendation scoring, ensuring more comprehensive and personalized suggestions.

We employed K-Means Clustering for segmenting the advisors. To effectively segment the advisors, we selected the following four features: Current AUM, Registered Years (Tenure), Current Production Segment, and Change in Production Segment (2022-2023). The Production Segment is MFS's categorization of advisors into Bronze, Silver, Gold, and Platinum statuses based on their performance. We transformed this into an ordinal value, assigning the values 0 (Bronze), 0.33 (Silver), 0.66 (Gold), and 1 (Platinum) to enable numerical analysis. The Change in Production Segment (2022-2023) feature captures the change in Production Segment, subtracting the 2022 value from the 2023 value.

To decide on the optimal number of clusters, we employed the elbow method. Based on this analysis, we determined that three clusters were appropriate for our data.





The clusters are described as follows:

Cluster 0:

- Highly Experienced (30 years)
- Low AUM ($1M)
- Bronze Segment
- Significant downgrade from previous year

Cluster 1:

- Experienced (21 years)
- High AUM ($8.9M)
- Mid-high Segment (Silver/Gold)
- Maintained position

Cluster 2:

- Less Experienced (12 years)
- Moderate AUM ($1.7M)
- Bronze Segment
- Slight downgrade from previous year

Evaluation Metrics

We focused on two metrics: recall@k and normalized discounted cumulative gain (NDCG@k), where k indicates the number of recommended items. These two metrics are useful for evaluating the proportion of topics correctly recommended and the underlying ranking, respectively. Both metrics are bounded between zero and one, with greater scores indicating better recommendations. Recall@k is specifically a classification metric, whereas NDCG@k obtains different results with continuous values, normalized between zero and one.

Both evaluation metrics can be directly applied to binarized topic counts, which takes on values zero and one. However, because flow attribution for topics can be negative, zero (missing), or positive, we must adjust these metrics. For best comparison, we decided not to penalize negative effects, instead calculating recall on positively attributed topics, and ignoring the magnitude of positive attribution. For NDCG, negatively attributed topics were ranked behind those with neutral or missing attribution, but attribution values were clipped between zero and one to ensure a positive numerator in the DCG formula, and therefore monotonically increasing DCG.

$$\mathbf{Recall@k} = \frac{\#\ good\ items\ in\ k}{\min(\ k, \#\ good\ items)}$$

$$\mathbf{DCG@k} = \sum_{k=1}^{K} \frac{2^{\mathbb{I}(k^{th}\ ordered\ item\ is\ good)} - 1}{\log(k+1)}$$

$$\mathbf{NDCG}: DCG\ normalized\ \epsilon\ [0, 1]$$

We highlighted recall@k with k equals one, three, and ten; and NDCG@10. We chose k equals one and three to evaluate the effectiveness of our best recommendation, plus the top three, which is the current plan for number shown to employees when implemented. We evaluated at k equals ten as well, to see how our rankings perform when looking deeper, which helps judge overall performance. Finally, NDCG@10 provides an indication of ranking strength. We thought k equals 10 to be effective, considering that it is large enough to judge ranking strength but much less than 73, the number of topics.

Model Performance

The recommendation system performs better than the baseline. For our baseline, we computed the probability distribution of each topic within the training set and randomly chose topics according to their respective probabilities. Unsurprisingly, the recommendation system performs much better, especially when comparing ranking strength and recall@3.

Model	Recall @ 3	Recall @ 10	NDCG @ 10
Baseline A	0.005	0.351	0.185
EASE + A	0.355	0.543	0.425
EASE + A + Clusters	0.376	0.562	0.444
Baseline B	0.101	0.304	0.156
EASE + B	0.263	0.386	0.246
EASE + B + Clusters	0.249	0.371	0.237

We presume that EASE + A has stronger recall and NDCG metrics compared to EASE + B for two reasons. First, recommending topics according to how they are currently chosen is, fundamentally, an easier task than recommending topics that we believe they would like. The recommendation system must learn how topic histories affect future topic choices, not how the outcomes of topic histories affect future topic outcomes. Second, it does not rely on the assumptions during creation of metric B: the relationship quality can be proxied and ignored (i.e. interactions directly lead to investment), and an interaction's influence decays exponentially for six months. It's likely that additional influences other than topics (market movements, interactions with other investment managers, etc.) result in flow changes, which remain uncaptured with metric B.

It's unsurprising that fitting an EASE model with each cluster does not improve results. Because EASE is a collaborative-filtering approach, it already learns client-specific representations. Though clustering has the benefit of introducing additional information outside of the client–topic matrix, reducing heterogeneity amongst each cluster, it reduces the total information EASE sees when training a model for each cluster separately. Clustering before collaborative-filtering isn't a principled approach, and we validate that training EASE on all users concurrently gives the best results.

Final Recommendation and Output Format

We suggest MFS incorporates our topic assignment and the EASE + B schema for the recommendation system. The necessary pipeline involves categorizing all previous interactions with their topics, then continually updating this for new interactions on a periodic basis, ideally daily. With the trained recommendation system, it is possible to retrieve recommendations on the fly if the client's topic history remains up to date.

Our stakeholders are unsure where the final recommendations will be visible. Some potential options are in C360 or a separate internal platform. Therefore, MFS has informed us that the final models and their outputs are a sufficient deliverable for our project. We decided to combine the results of our separate models into a single output format that displays all new relevant information. With a client's ID number as input, we plan to display their name and additional information:

1. Previous interactions: interaction ID, date, and associated topics
2. Recommended topics: topic name and a confidence score
3. Cluster information: cluster description and distance to center

These three sections encompass our technical contributions from the project.

Refactoring Code for Production

During our next weeks with MFS, after capstone deliverables, we will work on refactoring code into an object-oriented fashion such that it is suitable for production. We hope to complete this process before our departure to create an easy transition period and limited barriers to production.

Potential Use Cases

The developed tool for MFS Investment Management serves as a versatile solution with multiple applications across operational and strategic functions. By leveraging advanced data analytics and machine learning techniques, the tool enhances operational efficiency, client engagement, and strategic decision-making within the organization. The following key use cases highlight its potential impact:

Efficiency Enhancement for Sales Team:

The tool automates the summarization of client interactions into topic components and recommends topics based on historical data. By reducing manual data review and enhancing the speed of client interactions, it empowers sales teams to operate more efficiently, ultimately improving productivity.

Enhanced Client Personalization:

By accounting for a client's individual interaction history, the tool facilitates personalized client engagement strategies via topic recommendations. The sales team can tailor communication to align closely with individual client needs and preferences, fostering stronger relationships and therefore driving additional investment.

Client Segmentation and Targeting:

On its own, clustering financial advisors allows for better targeting with widespread marketing campaigns, particularly in email communications. This can be improved by including an advisor's topic preferences, as opposed to clustering in an attempt to improve the results of the recommendation system. By categorizing clients based on these preferences and engagement behaviors, MFS can tailor marketing efforts more effectively, enhancing retention strategies.

Potential Business Impact and Validation

We collaborated with the MFS team to discuss the potential business impact of our solution. We decided to quantify the increased efficiency for the approximately 70 external sales team employees that travel for in-person meetings roughly four times per week. We first calculated the baseline of their work currently. On average, these employees have a 40-hour work week, with four days dedicated to client meetings and one day dedicated to preparation and follow-up messages. We quantified roughly 25, one-hour meetings per week as a baseline, each with an approximate salary cost of $400. Upon viewing our results, the MFS team believed it could result in a 10% increase in meetings per week from efficiency gains alone, therefore resulting in an average of 27.5 meetings per week.

With these parameters, we estimate a 37.5% reduction in preparation time for external employees, resulting in $3.4 million of value generated from additional meetings.

The 37.5% reduction was calculated by determining the current time spent preparing for each meeting and comparing it with the projected time. Currently, sales professionals spend 8 hours preparing for 25 hours of meetings, or 0.32 hours per meeting. With our tool, the projected time becomes 5.5 hours for 27.5 hours of meetings, or 0.20 hours per meeting. The percent change equals 37.5%.

We calculated $3.4 million by multiplying the number of external sales professionals (70), the additional number of meetings per week (2.5), the number of working weeks per year (48), and the salary cost per meeting ($400). This value represents the increase from efficiency gains alone, not considering the possible generated value of improved topic selection. We do not have an estimate for the increase from more effective discussions.

Validation of our solution's impact will extend beyond initial implementation. We will create a plan for the MFS team to judge the behavior of the recommendation system while in production. For validation of the recommendation system, we advise a comparison of current topic selection behavior with that of the recommendation system to see if it makes intuitive sense. From an impact perspective, we advise collecting feedback from sales employees to measure efficiency and relationship improvements. These will influence long-term metrics, such as the number of advisors that MFS interacts with and net sales. This longitudinal approach will comprehensively evaluate the tool's effectiveness in driving business outcomes over time.

Future Work

Future work for our project can be broken up into technical and non-technical components. Technical components emphasize continuous improvement of technical methods and alignment with real-world behaviors, whereas non-technical methods emphasize the implementation, adoption, and integration into employee behaviors.

Technical Work

For topic assignment, it is possible to improve the percent classified of the current method by closely examining unclassified meeting notes and updating the keyword dictionary appropriately. However, it is likely a better use of resources to instead turn towards cutting-edge methods such as large language models (LLMs). Theoretically, LLMs could learn representations and locate relevant topics, as well as identify common themes that are currently unlabeled. This is being done alongside our project, particularly in retrieving non-topic information from the meeting notes.

The recommendation system can be improved by comparing EASE with other methods, selecting the best-performing method. Though we had our reasons for choosing EASE, there are a variety of other approaches that we have not yet tried. These may consider other approaches, such as content-based filtering or sequential recommendations. In any case, testing other algorithms, without changing the underlying data, may improve results.

A second way to improve the recommendation system is to introduce additional information. The most obvious variable to introduce is time, which is currently ignored during aggregation into the client–topic matrix. Sequential recommendations, or some metric adjustment to differentiate recent and old topics, could improve topic recommendations. Understanding how topics fit into a client's lifecycle, and incorporating frequency/scheduling of recommendations, can further automate the process of interacting with clients.

Non-Technical Work

The first step in future work would involve deployment to production of topic assignment and recommendation for sales team employees. This would allow employees to use our models. Ideally, the topic assignment model would run automatically per day, flagging each day's interactions with its associated topics. Therefore, recommended topics remain up to date.

Next, some form of feedback for the recommendation system would be useful. For example, allowing for data input to specify if the employee listened to the recommendation system's suggestions, or provided reasons for choosing their own topics instead. With this information, the MFS team can continue to tune the recommendation system while keeping its end users in mind.

On that note, improved data collection, particularly, in topic flagging and guessed sentiment, could deprecate the need for success metric B. At present, we defined success metric B to associate each topic with a positive or negative outcome to train the recommendation system. However, it would be much more effective to instead identify good and bad topics within each interaction as they occur, building out an improved dataset that no longer needs success metric B. MFS employees, upon concluding these meetings, could tag their own topics and identify how the client resonated with each topic, using either a negative/neutral/positive or one to five scale. Considering that our tool will save time in meeting preparation, spending time to flag topics after interactions does not increase total workload.

Appendix

1. https://arxiv.org/abs/1905.03375
2. https://huggingface.co/docs/transformers/en/model_doc/time_series_transformer