From Notes to Knowledge: Tailoring Future Client Interactions

Improving Preparation Efficiency of Sales Professionals

Project Members: Iris Brook and Xander Pero
Faculty Advisor: James Butler
MFS Team: Nadine Kawkabani, Brian Shaw, Suzet Nkwaya, Erin Haley

Project Context

At MFS, Sales team plays a pivotal role in driving client engagement and relationship management strategies

MFS Investment Management stands as a leading global asset manager. The firm oversees $615.1 billion in assets and employs over **350 sales professionals who sell investment products to financial institutions and advisors** through meetings (virtual and in-person), calls, and emails.

Primary Liaison: key interface between MFS and advisors

Client-Centric Approach: offer strategies tailored to client's needs

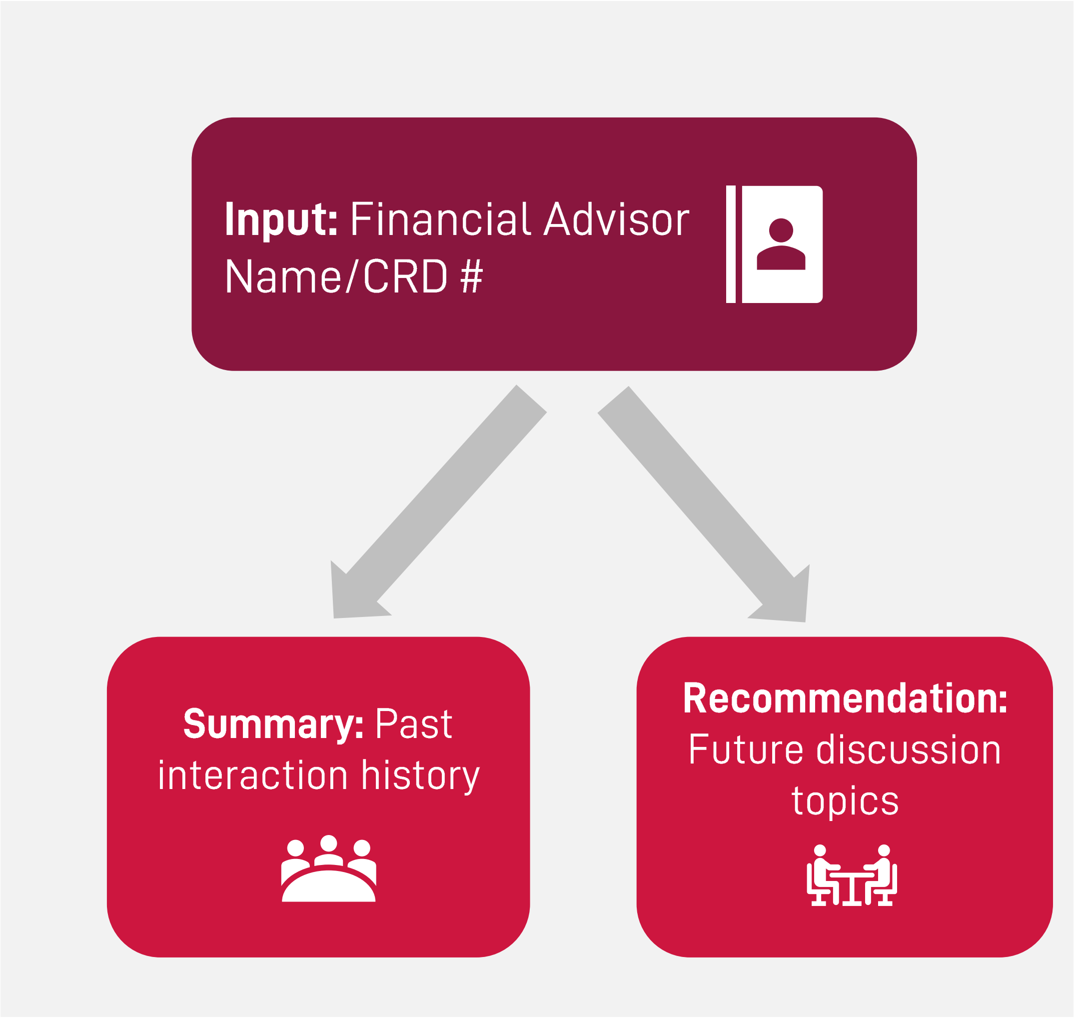
Proactive Engagement: provide support to maintain strong partnerships

70 External Sales Professionals **for** **130,000** Financial Advisors

Problem Statement & Goal

Problem Statement:

To boost **MFS sales team's preparation efficiency** with financial advisors, we want to develop a tool that provides **interaction history and recommends conversation topics that would drive sales.**



Value Generation Blueprint

Interaction → Relationship → Investment

Data:
- Date
- Note taken by sales team
- Client Info

No Data

Data:
- Date
- Amount
- Fund type
- Client Info

We will use post-interaction change in investment to proxy the client's relationship with MFS

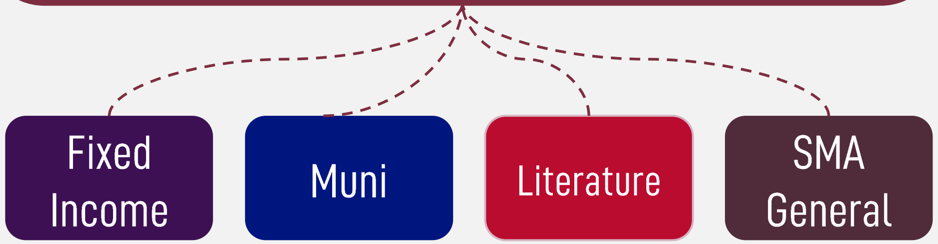
Interaction Summarization

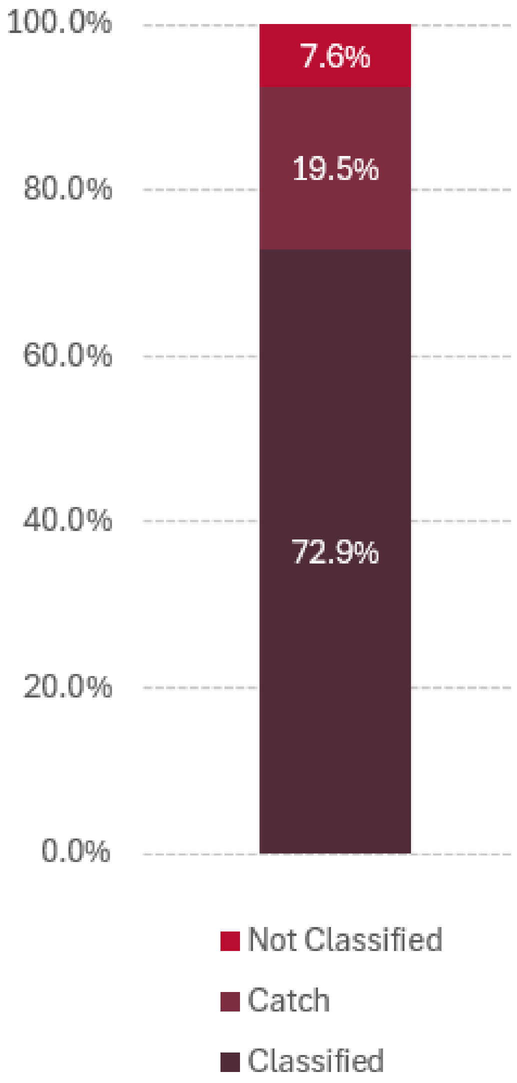
Topic Assignment Model:

- Analyzes meeting notes between sales employees and clients
- Assigns topics to meeting notes based on keyword presence and subject line content
- Catch – interactions without substance (out-of-office, scheduling, voicemail)

Example note with topic assignment:

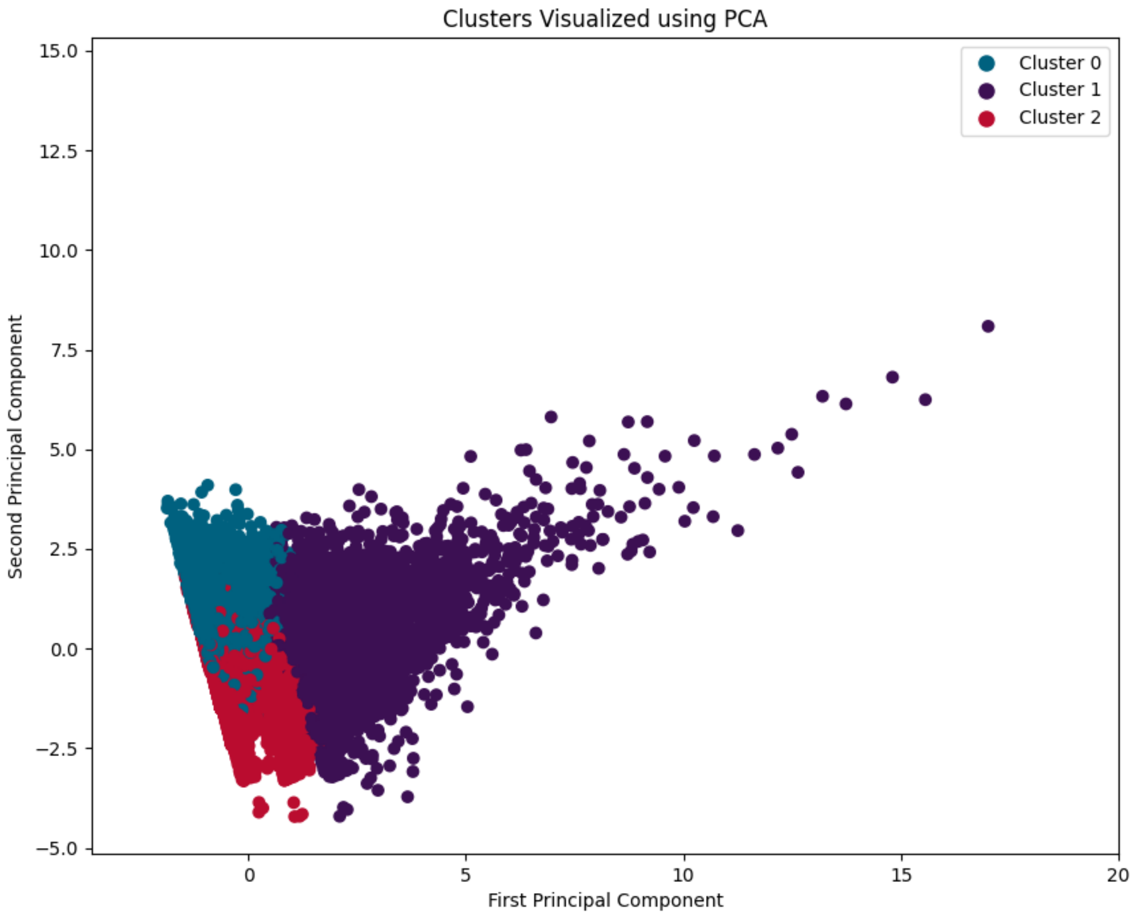
Great having lunch with Joe, always good to catch up with him. Discussed our thoughts on current credit quality and ppm. Sent over fact sheet on municipal bonds vs taxable bonds.





Clustering

Financial Advisors on Demographic and Client Data



Used K-means clustering to group financial advisors on four criteria:

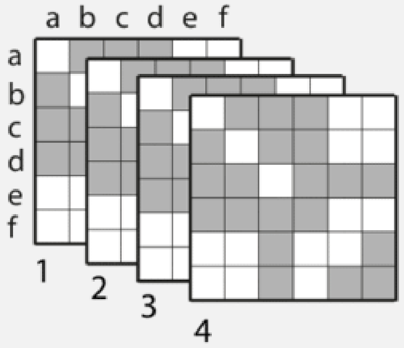
- Most recent holdings
- Registered Years (tenure)
- Current Production Segment
- Change in Production Segment from 2022-23

Cluster 0 – very experienced & low holdings
Cluster 1 – experienced & high holdings
Cluster 2 – less experienced & low holdings

Recommendation: Success Metrics

Tried out two different success metrics

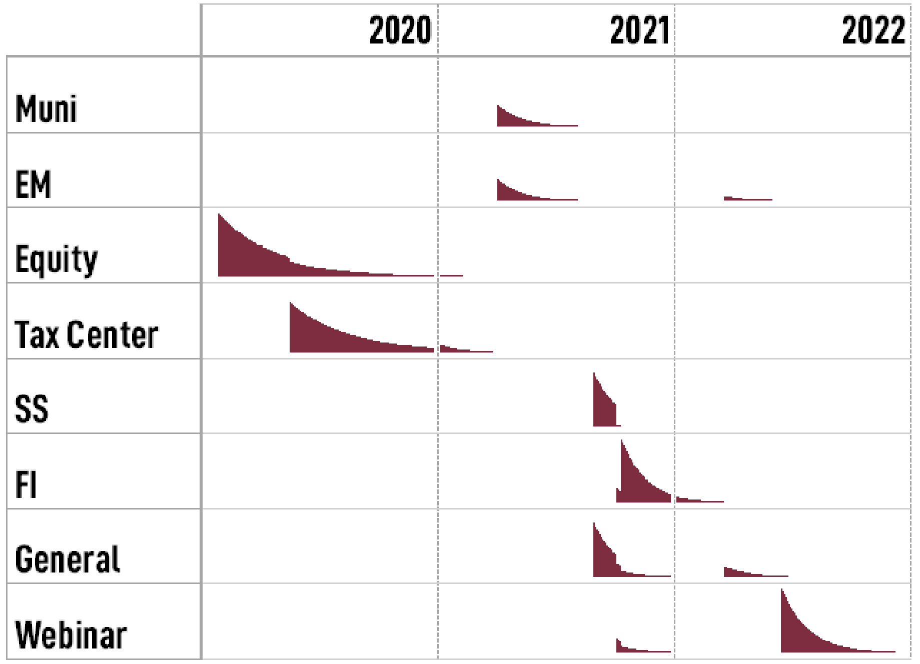
Success Metric A: Topic Counts



Data: 1 if specific topic was discussed with client on that day, 0 otherwise

Limitations: Does not consider financial advisor's perspective

Success Metric B: Post-interaction Investment



Data: Attributes a percentage of each client's flow to recently discussed topics

Benefit: Captures financial advisor perspective

Recommendation: Algorithm

A collaborative-filtering neighborhood-based approach with closed-form solution

EASE (Embarrassingly Shallow Autoencoders for Sparse Data) :

Why EASE?

- Less computation and training time than other collaborative-filtering approaches
- Single hyperparameter - less effort in tuning
- Competitively viable - similar results to other approaches

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

Recommendation: Model Preference

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

Recall@k: *Evaluates recommendation strength in top k*

Normalized Discounted Cumulative Gain (NDCG@k): *Takes into account order of ranking*

We recommend **EASE + B** (despite its slightly worse evaluation metrics) because it captures a topic's resulting impact, unlike metric A which does not.

Final Output

A dictionary of outputs for every client

Client	{MDM ID, Name, History, Recommendations, Cluster}

5 Most Recent Interactions

Interaction ID	Date	Topics
247653	7/29/23	Muni, Tax Center
985375	7/30/23	EM Equity
1034876	9/4/23	Missed Call
1074563	10/22/23	Tax Center
1105487	12/15/23	Share Class

Topic	Score
Fixed Income	0.82
Share Class	0.76
EM Equity	0.34

3 Topic Recommendations

Cluster:	1
Description	• Experienced • High AUM • Silver-Gold • Maintained position
Distance to Center	0.37

Cluster Information

Business Impact & Next Steps

37.5%

time back for sales employees

$3.4 million

of generated value for MFS annually

Future Work:

- Implement tool & collect feedback on recommendation quality
- Compare EASE to more powerful algorithms that include temporal factors or order of interactions

Thank You.

Questions?

Appendix

Appendix: Recall@k & NDCG@k Formulas

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

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

Normalized Discounted Cumulative Gain (NDCG):

Takes into account order of ranking

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

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

Appendix: EASE Algorithm

A collaborative-filtering neighborhood-based approach with closed-form solution

Optimization Formulation

$$\min_{B} \quad ||X - XB||_F^2 + \lambda \cdot ||B||_F^2$$
$$\text{s.t.} \quad \operatorname{diag}(B) = 0$$

- X: Client-Topic data matrix
- B: learned weight matrix
- λ : Regularization hyperparameter

Closed-Form Solution

$$\hat{B}_{i,j} = \begin{cases} 0 & \text{if } i = j \\ -\frac{\hat{P}_{ij}}{\hat{P}_{jj}} & \text{otherwise.} \end{cases}$$

- $\hat{P}$: $(X^{\top}X + \lambda I)^{-1}$
- $\hat{B}$: learned weight matrix

Training Algorithm

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
```

Appendix: Business Impact Calculations

37.5%

time back for sales employees

$3.4 million

of generated value for MFS annually

Pre-Tool: 25 meetings/week

With Tool: +10% = 27.5 meetings/week

Meeting Salary Cost: $400

Pre-Tool: 8/25 = 0.32 hr/meeting

With Tool: 5.5/27.5 = 0.2 hr/meeting

= 37.5% reduction in time

(70 people) x (48 weeks) x (2.5) x ($400)

= $3.4 million