Module #2
Needs-based recommendation systems, collaborative filtering, Next Best Actions (NBA)

Raffaele Zenti (raffaele.zenti@wealthype.it)

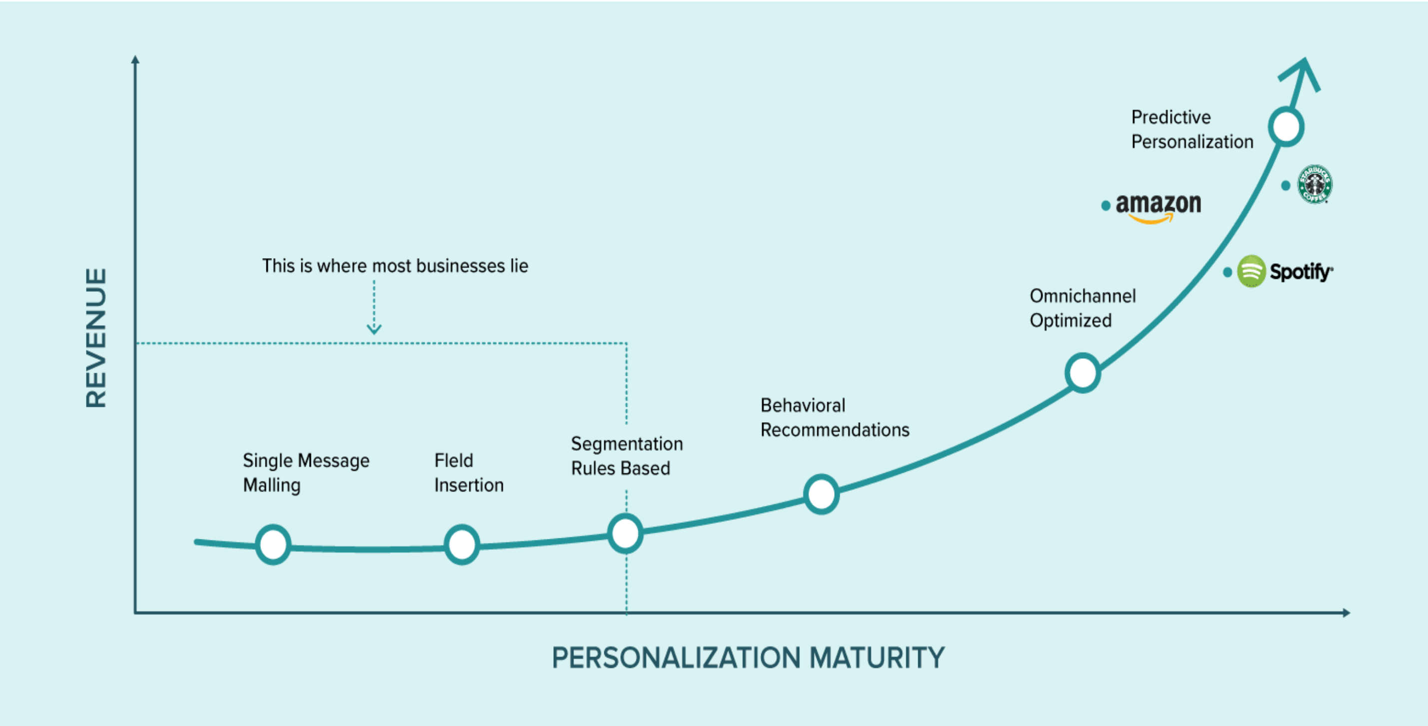
Co-Founder, COO, Chief Data Scientist

Whealthype-AI SpA

The key to financial personalization

Hyper-personalization

Netflixing” of financial services



«In 2030, up to ***80 percent of new wealth-management clients will want to access advice in a Netflix-style*** *model—that is, data-driven, hyper-personalized, continuous, and potentially, by subscription.»*

Source: McKinsey & C

Needs/goals are key to targeting clients

- After all, we are still monkeys: most of us don't really understand financial and insurance products (so we don't buy…)
- But we all have real needs – and we understand them, e.g.:
  - protect ourself, our family, our things → insurance
  - buy things → payment tools & services
  - save for future consumption → savings & investments
  - anticipate future consumption, or investments → borrowing

We tend buy what we need: thus, needs are a good starting point for recommending financial products and services in a personalized way

Another good reason to estimate client's needs: the law

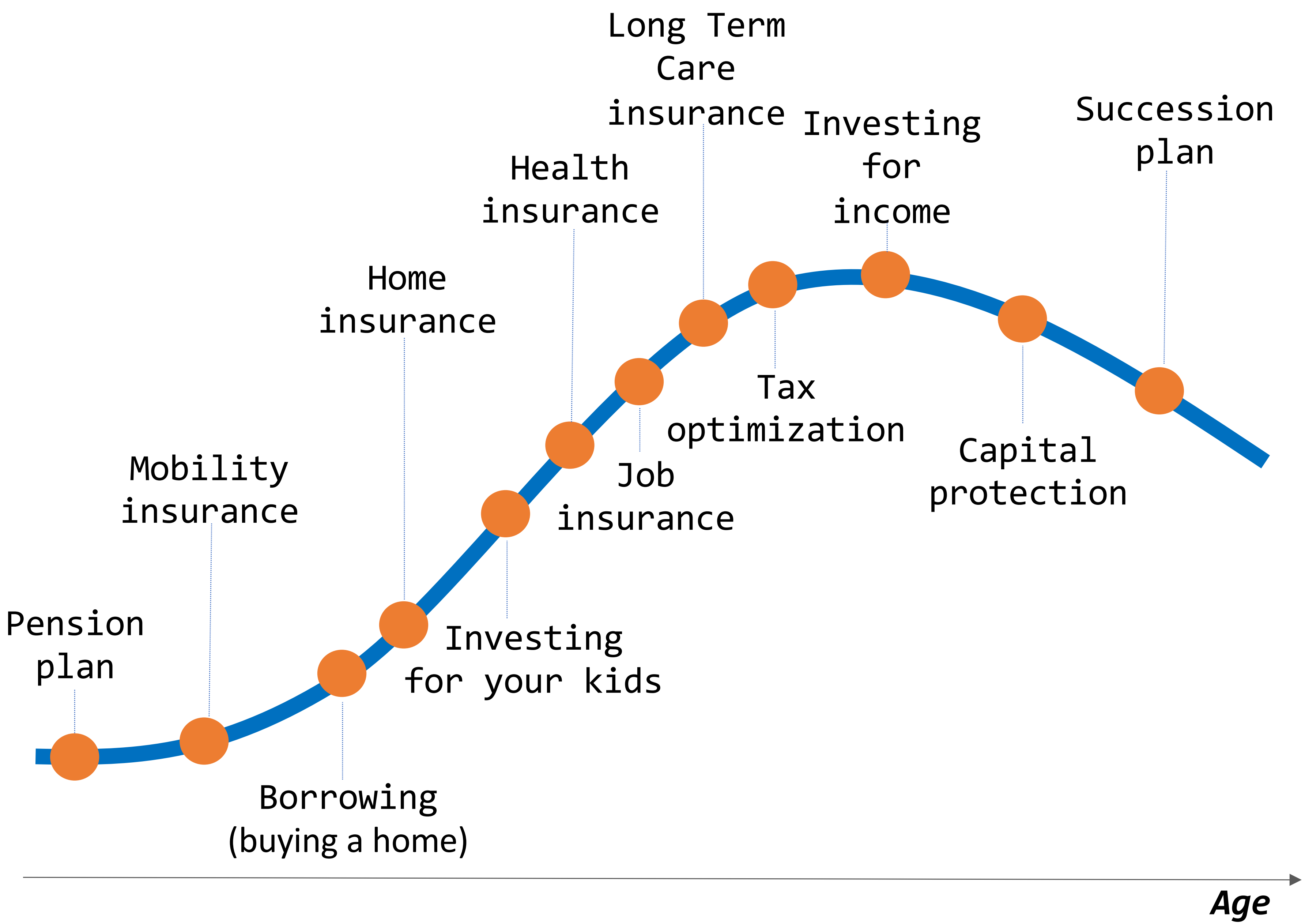
- MIFID/IDD: coherence between needs/goals and financial/insurance products sold to clients

- Key information about clients must be collected by the use of a MIFID/IDD questionnaire → Data → a lot of detailed information collected through MIFID/IDD questionnaires can be crunched by algorithms

- Basically, you get a broad survey for free – that's the reason why MIFID/IDD questionnaires should be properly prepared

"Make a virtue out of necessity"

Financial needs: “the theory”



The reality of financial & insurance needs

- Not everybody will start a family at 30…
- Maybe at 72 not everybody is willing to plan her inheritance process, maybe is getting ready for a marathon or sailing around the world
- What about if at 50 you have 2 divorces and 2 maintenance allowances?
- Maybe at 35 someone faces a big recession, is fired, and cannot buy a home
- …

Financial needs change over time following our random life

And our random lifes are not all equal

Financial needs change overtime

How? Ask data!

Need(i, t) = f(client situation(t), context(t))

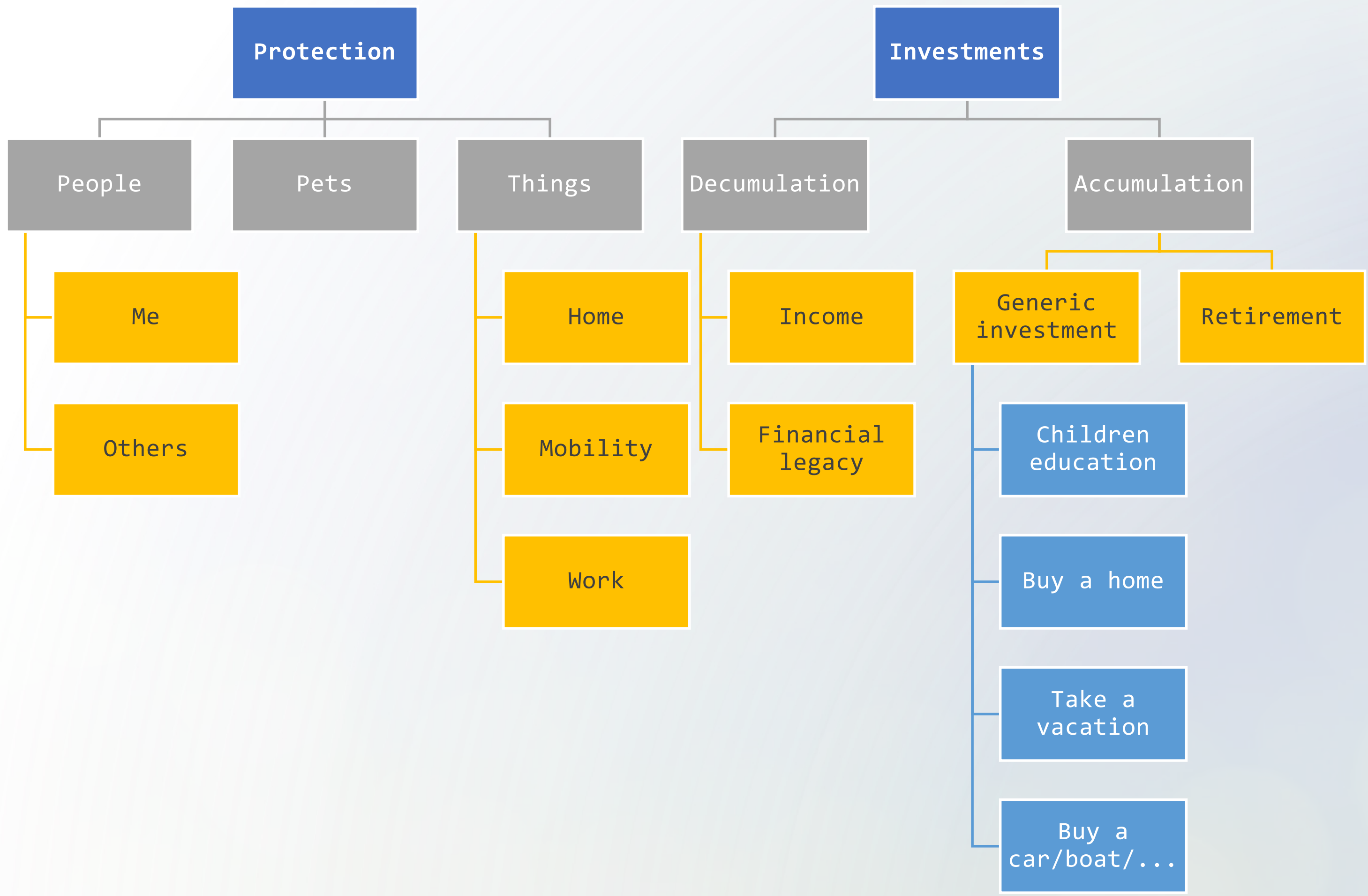
That is (...smell of ML):

$$\boldsymbol{Y = f(x_1, x_2, x_3, \dots)}$$

$\boldsymbol{Y}$ = responses = Need

$\boldsymbol{x_1, x_2, x_3, \dots = X}$ = features = client situation, context

Needs & goals, an example for an insurance company with a wealth management business



Let's look at the data

Estimating needs: a Supervised ML problem

- A client might have/not have a given financial need (or goal)
- Each client might have more needs
- Needs can be satisfied by financial products

I could teach a ML algorithm to recognize presence/absence of needs

It's a typical classification problem

(it might be a regression, too)

And what about the Ys? Welcome to the real world!

- In a classification problem we teach an algorithm to put labels
- But... Where are the labels?
- Who is able to say: «Client A has need Z»?
- **Needs are not observable!**
- Hence, we have our Xs but we are not sure about the Ys → We have a problem...

«I'm Mr Wolf, I Solve Problems»

- Basically, two ways:

  1. Straight - explicit labels

     «Client A has/has not need Z»

  2. Less straight - implicit labels

Case 1: explicit labels =
= a human being creates the Ys

- She puts:

  Y = 1 if client(i) has need(j)

  Y = 0 otherwise

- Quite common in image recognition
- If the human-labeler is reliable $\rightarrow$ very good
- But: financial needs are not easy to spot – they are not cats/dogs/pedestrians/etc
- So: you need investment/insurance/banking experts $\rightarrow$ expensive

Case 2: Implicit labels =
= Ys inferred from <u>expert behaviour</u>

- Learn from those who should know if a given client has a given need: financial advisors

- If an advisor sells a financial product that satisfies a given need, then probably it was in order to satisfy a need

- Thus:

  Y = 1 if client(i) owns a product that satisfies need(j)

  Y = 0 otherwise

- If the human sells products that maximize <u>HER OWN profits</u> → the algorithm will learn exactly that process (do you remember all these fancy talks about AI and ethics?) ➜ Solutions:
  - filtering experts and their behaviors, doing «expert-picking»
  - using a priori information (Bayesian models)/combine different models (see Bayesian Model Averaging)

One-vs-All models
Vs
Multiclass models

- **One-vs-All (Binary classification, Binomial)**:
- 1 need ↔ 1 model
- As many models as there are needs (10 needs ↔ 10 models, 30 ↔ needs, etc)

- **True multiclass (Multiclass, Multinomial)**:
- N needs ↔ 1 model
- One single big model (10 needs ↔ 1 model, 100 needs ↔ 1 model, etc)

  → often more complex = might be less robust (see Occam’s Razor)

From needs to recommendation (NBA): finding the best matching

- **Content-based filtering** - knowledge-based methods that rely mostly on the domain-knowledge
- **Case-based recommender systems** - apply case-based reasoning (CBR) that solves the recommendation problem based on similar cases

Other approaches - Collaborative filtering: learning from similar situations/guys

Using Singular Value Decomposition or an Autoencoder to Build a Recommender System

Method based on latent variables

Often it does not go to the heart of the matter: it is unable to manage complex situations (for example, in recommending financial products, it is necessary to take into account many factors, primarily regulatory constraints).

Coding session starts