Fintech final project
A.Y. 2021/2022

Teo Bucci
Filippo Cipriani
Gabriele Corbo
Davide Fabroni
Marco Lucchini

Estimating Clients' Needs

Crafting an easy-to-use tool to recommend products to clients.

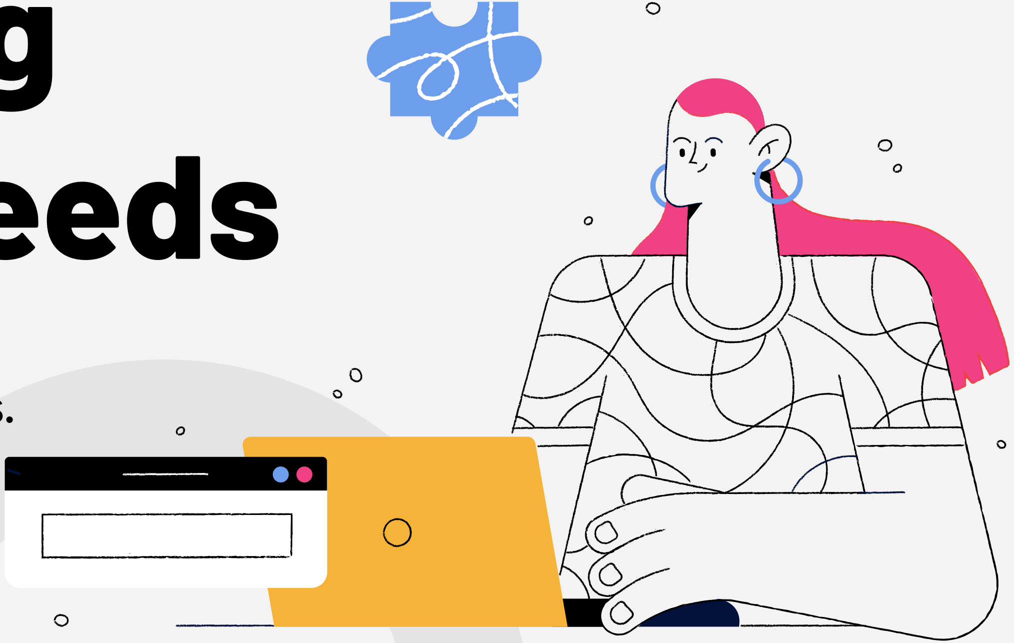


Table of contents

01
Our Goal
How we envisioned the web-app for products recommendations

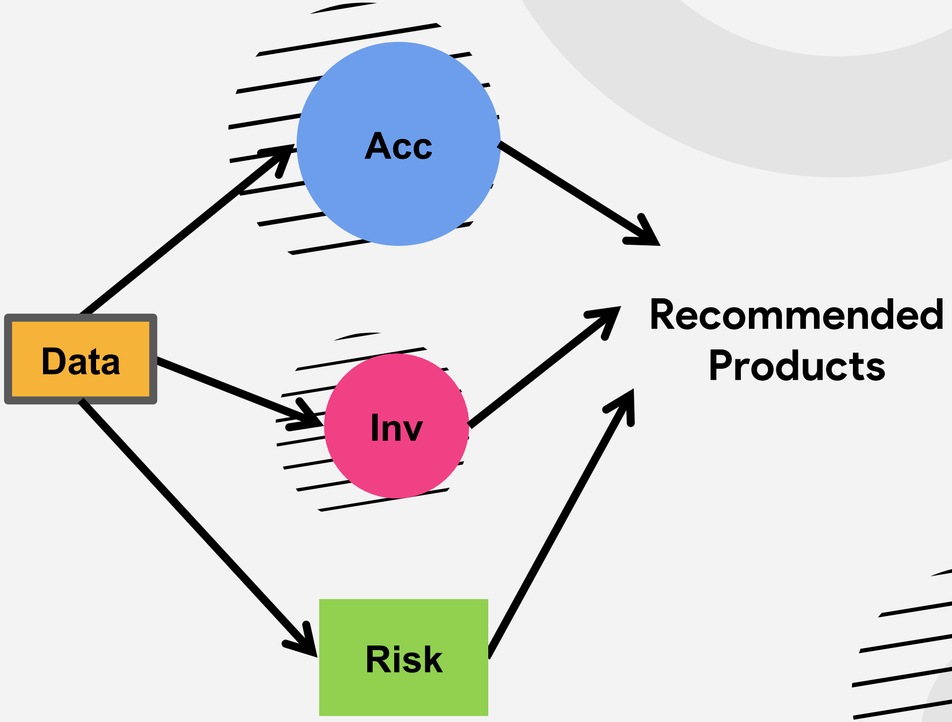
02
The workflow
The tools used for app development

03
The dataset
Clients and products data structure

04
Estimating risk
How Risk Propensity is factored into the model

05
Segmenting clients
Building a classification model for recommending certain types of products

06
Deployment
Try the deployed app and see the resources



01

Our goal

A web-app for recommending financial products to clients:

- Easy-to-use and deploy
- Minimal interface
- Factor-in product type and client's risk propensity

02
The workflow

We followed a multi-tool approach, porting models to exploit each software's development strengths

MATLAB

R

Used for data exploration and model selection

- Fast prototyping
- "Push-the-button" model exploration
- Useful pre-implemented functions

python™

Quickly encode the selected MATLAB model

- Open architecture
- Easy-to-use web development modules

Streamlit

Quick development and deployment of a web-app with little to no knowledge required

State-of-the-art Data Science tools

Steps

1. Exploiting MATLAB exploration functionality to find and then build in R the best linear model to estimate the Risk propensity of a client.

2. Build a (supervised) classification model in MATLAB to predict the preferred products based on clients data and on the risk calculated with the linear model.

3. Porting the model in Python.

4. Development and deployment of an interactive web-app using the Streamlit module.

03 The dataset

Needs: clients data

Needs | Products | Data

5000x11 table

	1 ID	2 Age	3 Gender	4 FamilyMembers	5 FinancialEducation	6 RiskPropensity	7 Income	8 Wealth	9 IncomeInvestment	10 AccumulationInvestment	11 FinancialStatus
1	1	60	0	2	0.2287	0.2334	68.1815	53.2601	0	1	0.9091
2	2	78	0	2	0.3589	0.1709	21.8076	135.5500	1	0	1.7620
3	3	33	1	2	0.3175	0.2497	23.2527	66.3037	0	1	1.3317
4	4	69	1	4	0.7677	0.6546	166.1890	404.9977	1	1	4.6091
5	5	58	0	3	0.4297	0.3490	21.1867	58.9119	0	0	1.7516
6	6	42	1	3	0.4798	0.5144	89.9044	78.6230	0	1	2.0940
7	7	50	0	2	0.4855	0.3783	63.9291	53.4008	0	1	1.9313
8	8	59	1	2	0.3743	0.2161	131.4302	100.6633	0	1	1.7263
9	9	81	0	3	0.1951	0.1775	95.1564	131.8780	1	1	0.9525
10	10	71	0	2	0.5773	0.4542	104.6945	199.4728	1	0	3.0574
11	11	45	0	3	0.4727	0.5881	106.7780	84.5515	0	1	2.0975
12	12	81	1	2	0.2525	0.3165	55.5094	87.9851	0	0	1.1307
13	13	67	1	2	0.3245	0.3419	109.9302	80.2690	0	1	1.4233
14	14	54	1	2	0.6996	0.6805	74.4971	367.1692	1	1	4.1316
15	15	61	1	3	0.4173	0.3372	81.0121	159.4875	1	1	2.1165
16	16	53	1	2	0.4270	0.2207	45.2375	48.0160	0	1	1.6533

We have the following information for each client:

- Age
- Gender
- Family Members
- Wealth/Income
- Financial Education
- Risk Propensity
- Desired type of product

Products: list of possible products recommendations

Products are identified upon

- Type (inv/acc)
- Risk level

Needs | Products

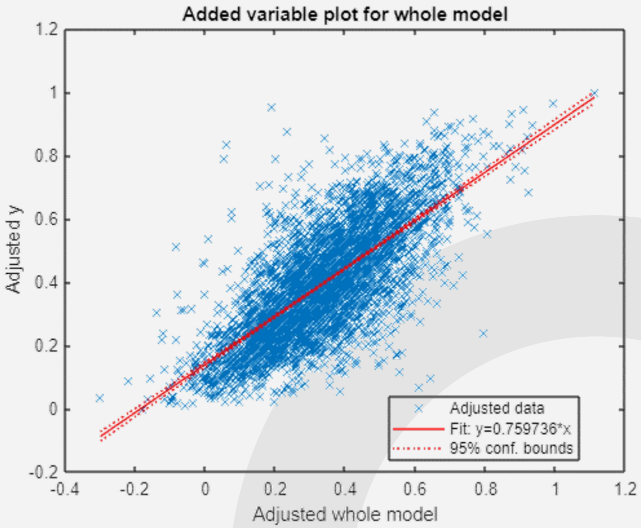
11x3 table

	1 ID	2 Type	3 Risk
1	1	1	0.5500
2	2	0	0.3000
3	3	0	0.1200
4	4	0	0.4400
5	5	1	0.4100
6	6	1	0.3600
7	7	1	0.7500
8	8	1	0.4800
9	9	1	0.2700
10	10	0	0.1300
11	11	1	0.8800

04 Risk propensity

We wanted to factor out the Risk from the data that has to be asked to the client.

Initially, we looked at international regulators such as MiFID to try to build a deterministic index: lacking some information, we turned to a supervised regression model.

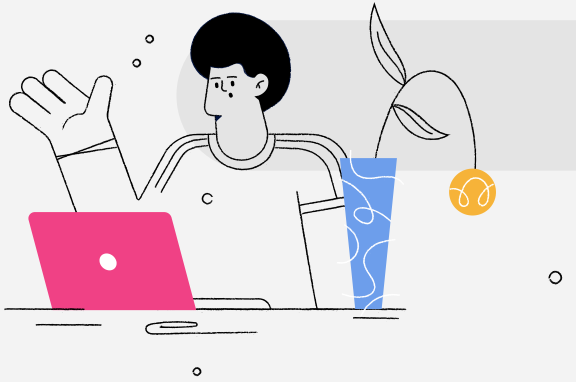


Using the Regression Learner we found that Linear Model was the best one. We jumped to R to generate it and studied interactions between the regressors.

Adding the interaction **FinancialEducation * log(Wealth)** (a sort of financial status) significantly increased the variability explained by the model.

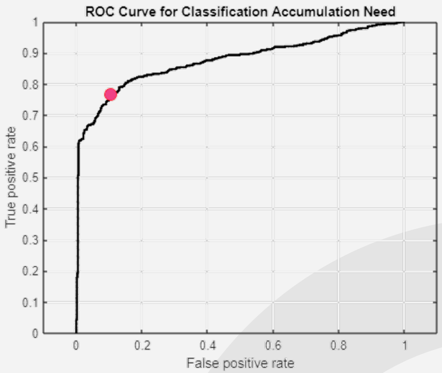
R^2 of the model = 0.52

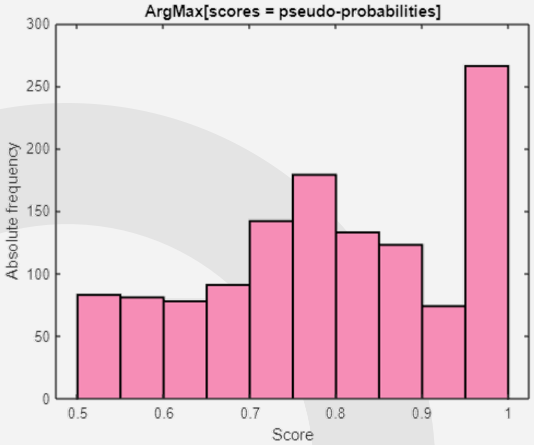
05 Clients segmentation



Using the Classification Learner, we looked for the best model in predicting the appropriate type of product, accumulation or investment.

Turns out the models that provide the best results are Bagged Trees for Accumulation and LogitBoost for Income.





06 Deployment

Once we had our two models, we thought that the Python library Streamlit would be the best one to deliver an easy to use tool, that requires no installation and has no compatibility issues between platforms.

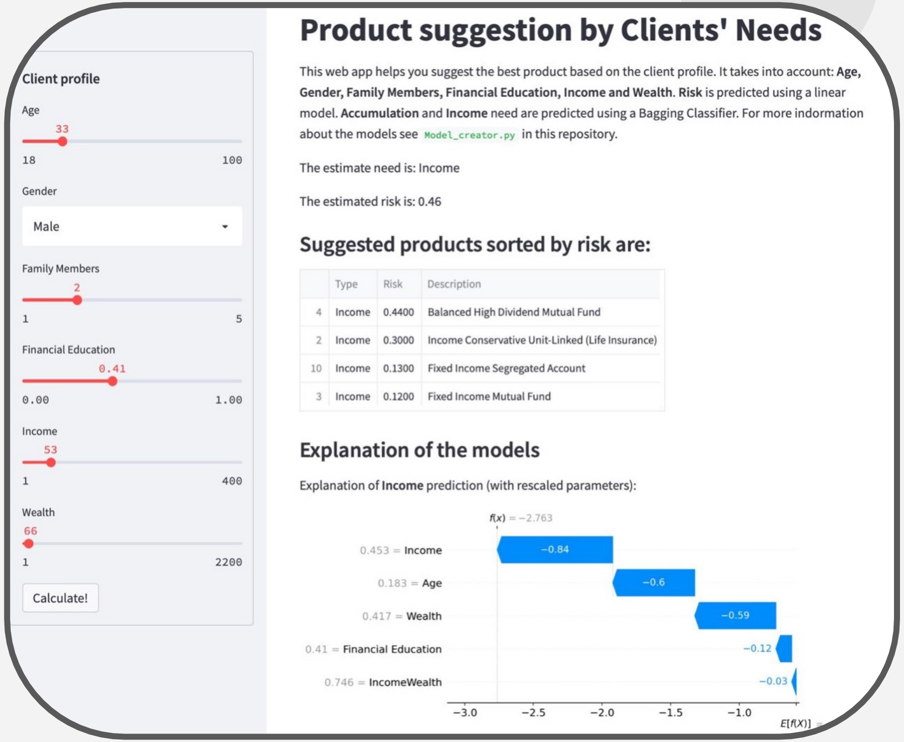
In a few lines of code we brought to Python the Linear Model and the Bagged Tree algorithm

Try it out!

The web-app is online and ready to use.

Use the menu and sliders on the left to input client data, press calculate to display the suitable products!

https://share.streamlit.io/marcolucchini/fintech-project/main/main_Streamlit.py



Conclusions

Further developments

- Add a MiFID-based survey to build a more reliable risk estimation model

References

- docs.streamlit.io
- it.mathworks.com/help/stats/classificationlearner-app.html
- r-project.org
- Hair, J. F., Ringle, C. M., & Sarstedt, M. (2011). PLS-SEM: Indeed a silver bullet. Journal of Marketing theory and Practice, 19(2), 139-152.

Thank you