Client segmentation and product recommendation

An ML tool in support of Bank’s decision making process

A work by Luca Amadori, Margherita Bencini, Andrea Catelli, Rodolfo Coppola, Stefano Corti

Project pipeline

Data analysis

We analyze the customer dataset by examining relationships and variable distributions. Then, we look at the product dataset and **compare the risk distribution** of the products with the customers' risk propensity distribution in order to assess the variety and suitability of the bank's offering.

Product creation

We assess the possibility of improving the bank's catalogue of products, and thus, we try to **create additional products** to better match the needs of our clients. The creation process is based on the replication of different indexes through the use of futures derivatives.

ML

We apply machine learning to **predict customers' investment propensity** via income and accumulation strategies, enhancing performance through feature engineering and hyperparameters tuning. To quantify the financial impact, we introduce a **Business Impact Score** (BIS).

Recommender

We build up a recommendation system which tries to **match the products** in the bank's catalogue with the clients' risk propensities based on the forecasts made in the previous step.

Data analysis

Age

Family members

Risk propensity

Wealth

Gender

Financial education

Income

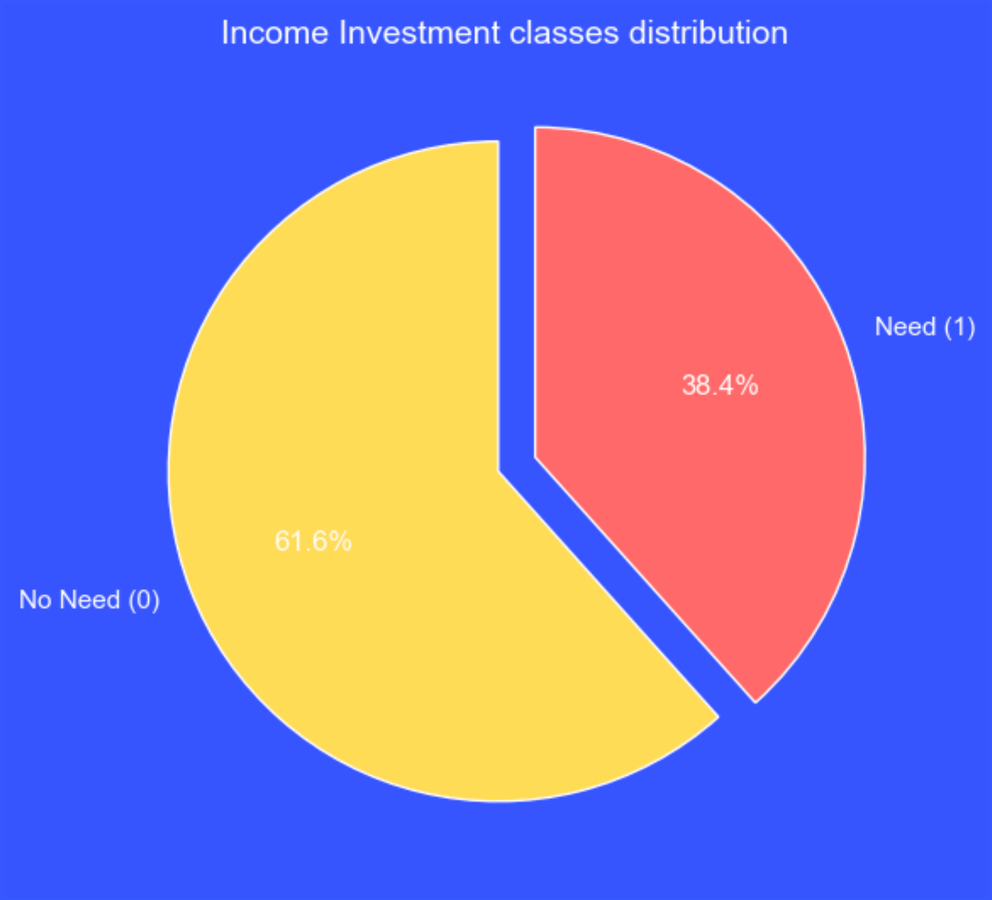
Income Investment

A boolean variable that indicates a customer's propensity for income oriented investments - financial products designed to provide regular payouts

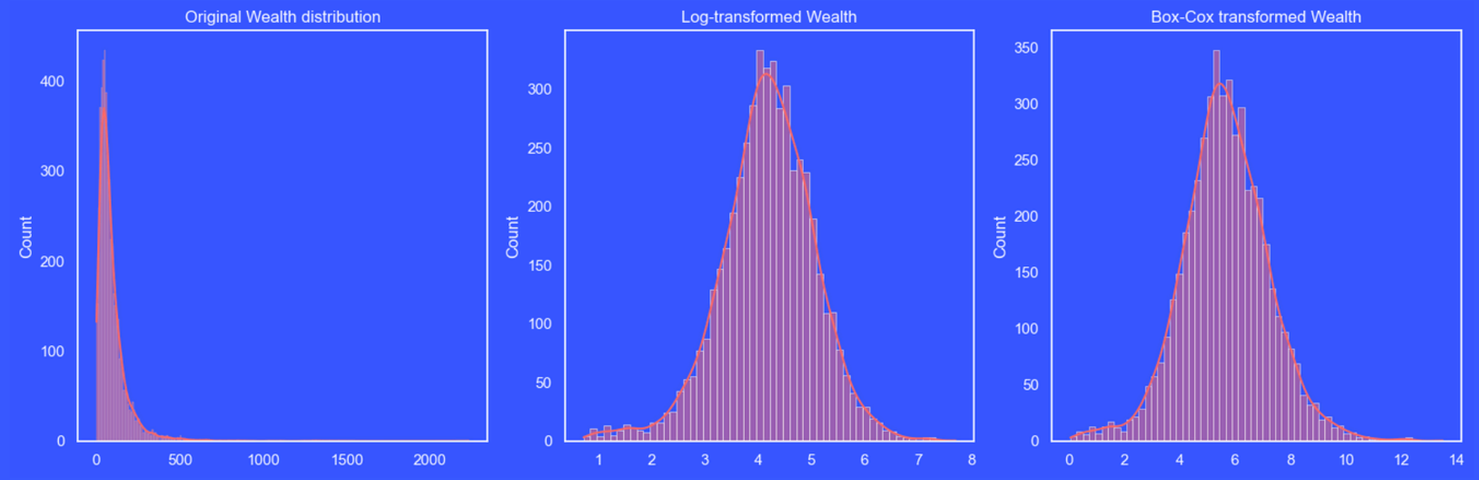
A boolean variable that reflects a propensity for accumulation oriented investments, where any returns are reinvested rather than paid out

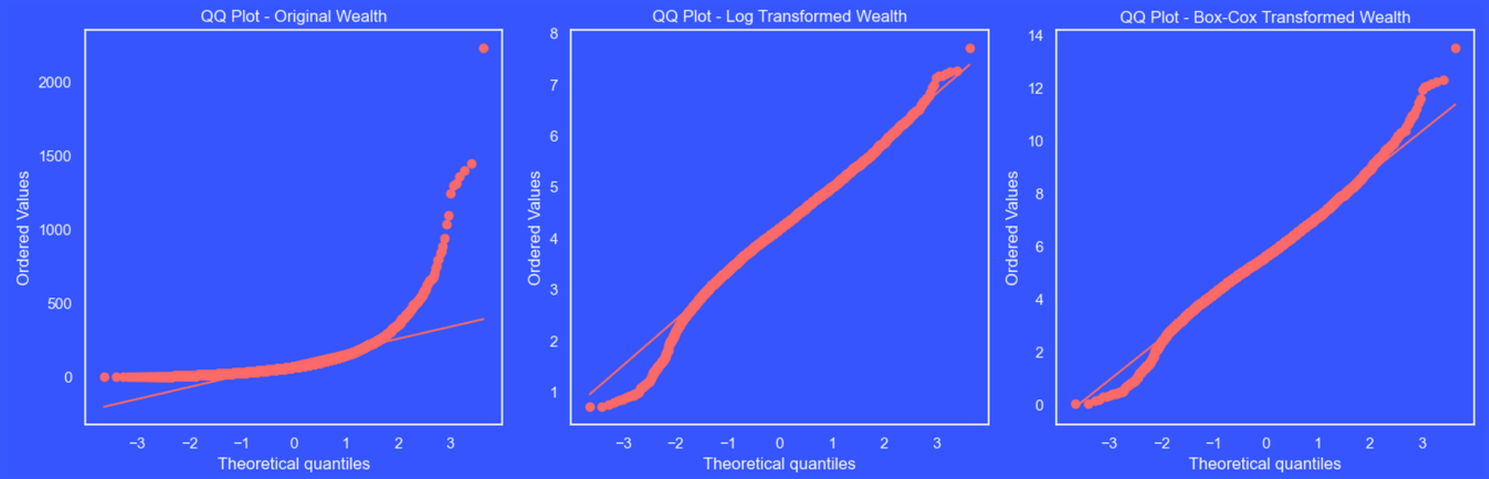
Accumulation Investment

Our dataset has **11 different products**, classified both in terms of their risk level and divided in Accumulation and Income. They space from a Flexible Aggressive ULIP to a Fixed Income Mutual Fund.

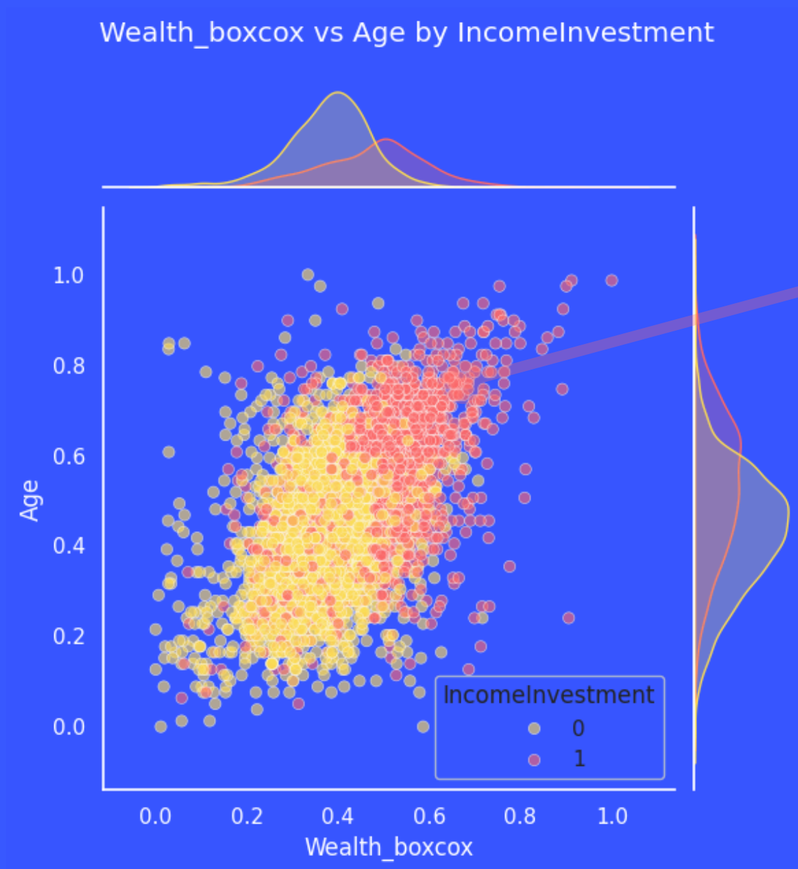


- The pie chart shows a moderate class imbalance in the Income Investment variable: while not extremely skewed, the **Need** class is underrepresented compared to the **No Need** one.
- It’s crucial to consider this aspect, as the imbalance may lead classification models to favour the majority class, reducing sensitivity to the minority group.

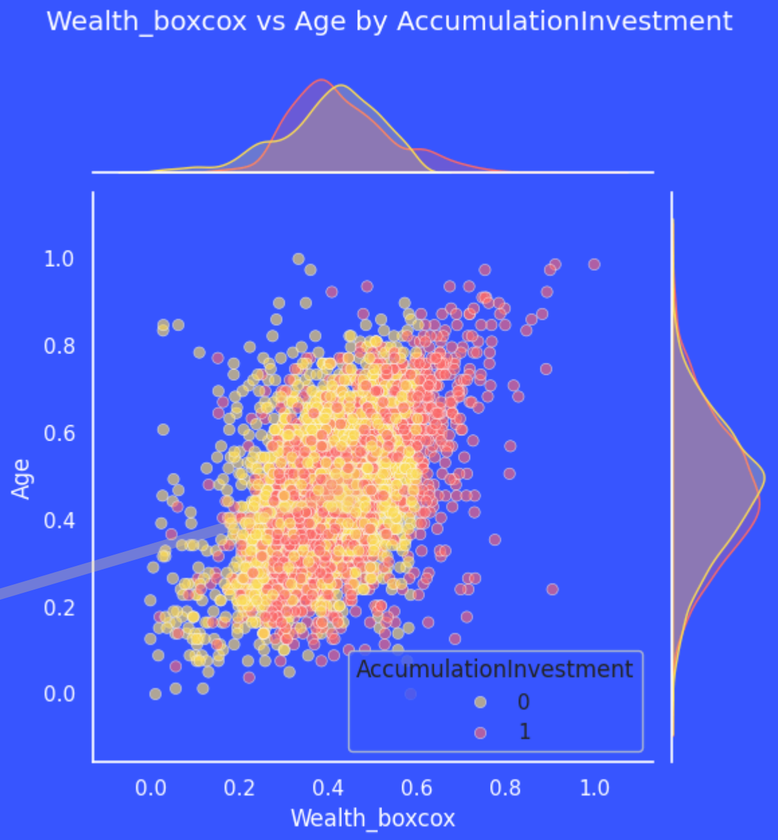




- Focusing our attention on **Wealth** feature, we noticed a highly skewed distribution which could be problematic for models' performances: we applied **Logarithmic** and **Box-Cox transformations** in order to reach as much as possible gaussianity.
- As QQ-plots results suggested, we added the Box-Cox transformed **Wealth** feature to our dataset.
- We performed the **same analysis** on **Income** variable, getting very similar results.

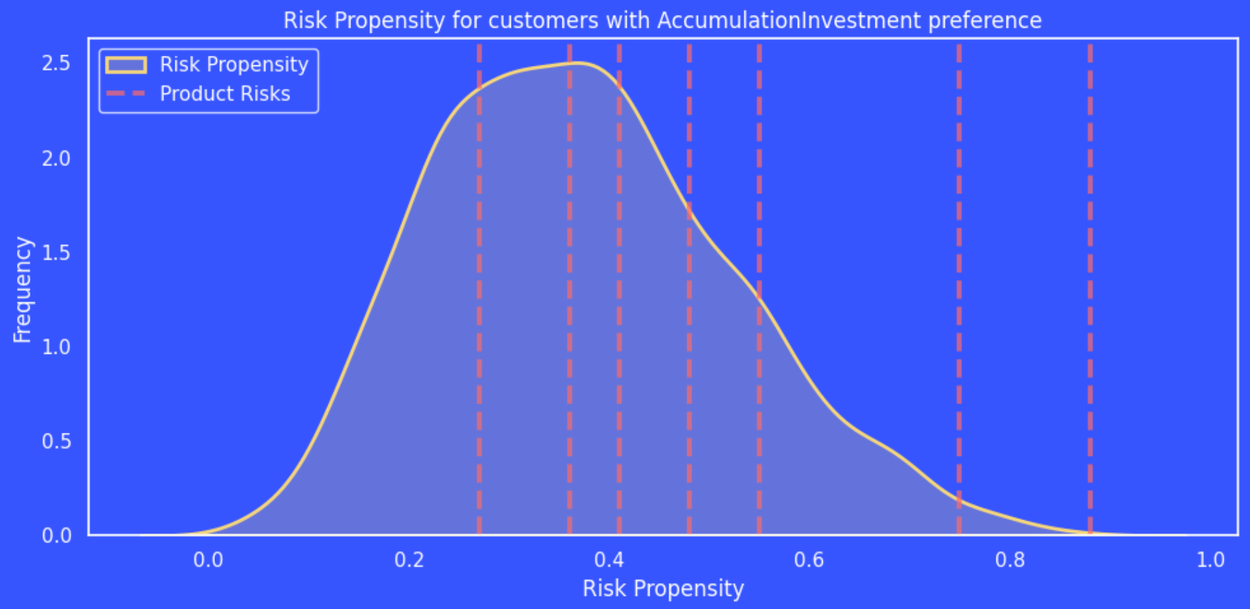


The plot clearly shows how **Income** investment prone people are generally **older and wealthier**: this could be good news for modelling purposes.



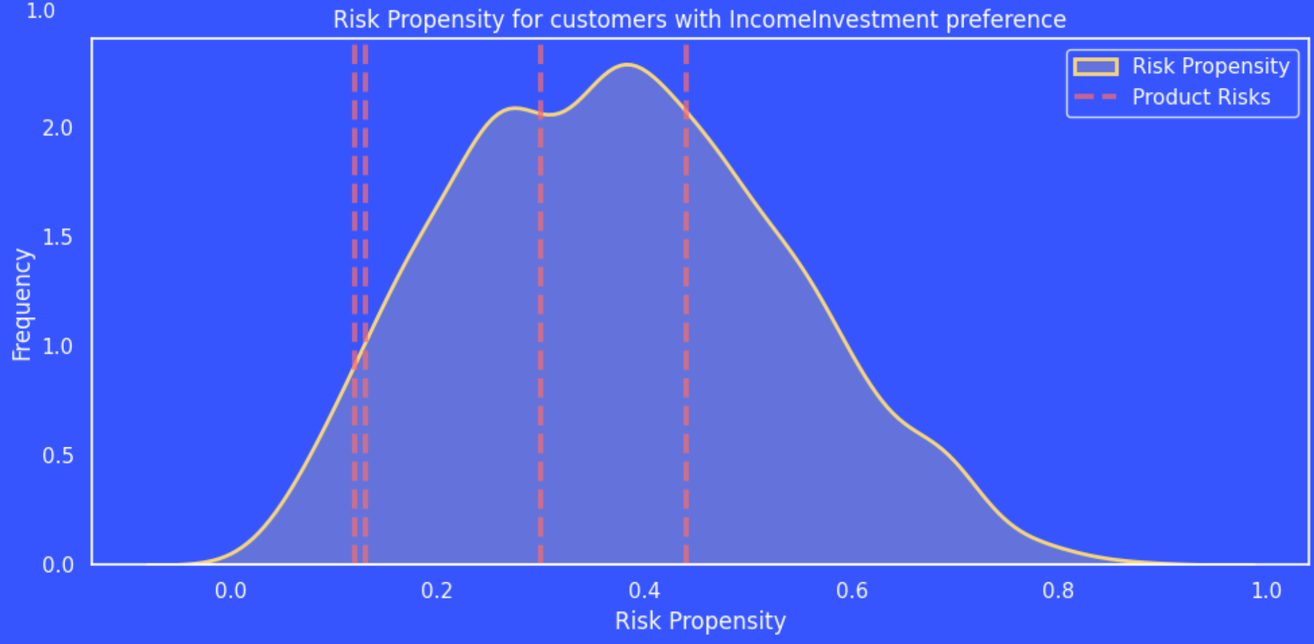
The joint distribution of **Age** and Transformed **Wealth**, classified by the **Accumulation propensity**, reveals a more heterogeneous dataset, with no such clear indications as in the case of **Income** propensity.

Products creation



- Regarding **Accumulation**-type products, the bank's catalogue presents scarcity of lower risk possibilities, possibly leaving lots of clients unsatisfied.
- We decided to **create some "safe" investment possibilities**, expanding the offering to customers.

- Regarding **Income**-type products, the bank's catalogue lacks offerings at the **extremes of the risk spectrum,** both very low and very high.
- We tried to **add an extreme-risk investment possibility**, enabling product recommendations to be more aligned with actual customer needs.



Address the **left tail** of clients' risk distribution for **Accumulation preference**

Extreme lower end of the tail

Time deposit risk 0.02

Upper portion of the tail

Synthetic product via replication risk 0.15

Address the **extremes** (high and low) of clients' risk distribution for **Income preference**

Low level of risk

Time deposit risk 0.02

High level of risk

Attempted to create a product with high-level of risk

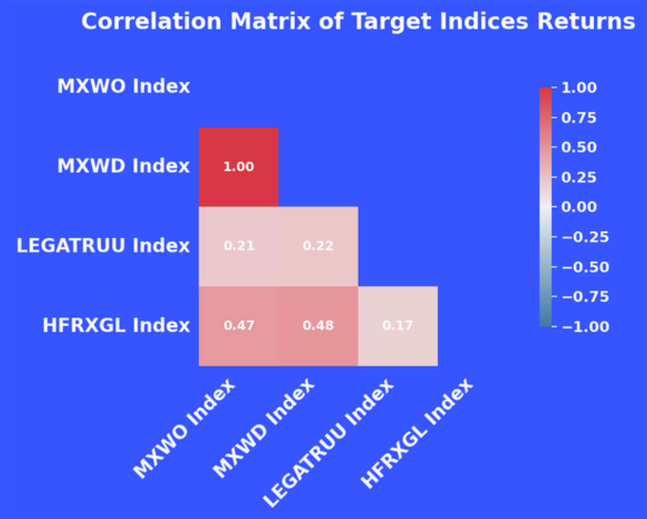
$$SRRI = \sqrt{\frac{\Delta}{T-1}\sum_{i=1}^{n}(r_i - \bar{r})^2}$$

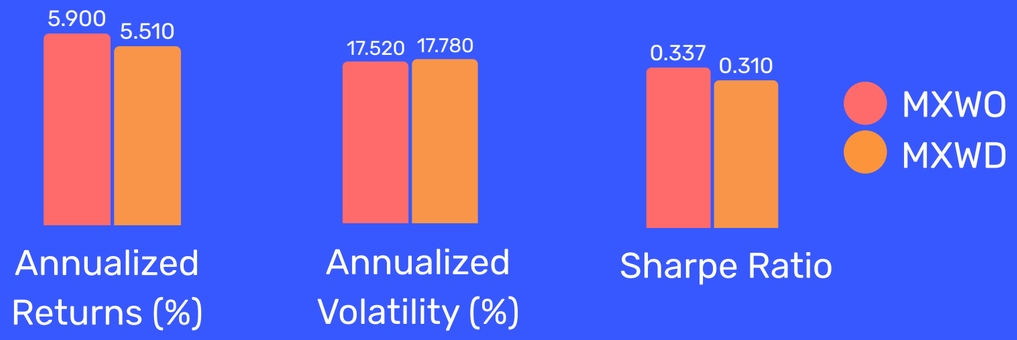
$$\text{where} \quad \bar{r} = \frac{1}{n}\sum_{i=1}^{n} r_i, \quad T = 260, \quad \Delta = 52$$

The synthetic product replicates a target index using cheap, liquid, and easy-to-trade **futures**. Risk is measured using the **Synthetic Risk and Reward Indicator (SRRI)**, following CESR guidelines based on five years of weekly returns to compute annualized volatility. Among portfolios with SRRI in the **desired risk range**, the one with the **highest annualized return** is selected.

Construction of the target index:

MXWO | MXWD | LEGATRUU | HFRXGL





We excluded the MXWD due to its **perfect correlation** with the MXWO, which had a **slightly higher annualized return** and lower volatility.

For the Accumulation-type product, we obtained a target portfolio with SRRI equal to 0.15

Weights of target portfolio
Accumulation product (SRRI = 0.15)

MXWO	0.983
LEGATRUU	-0.436
HFRXGL	0.453

Statistic of target portfolio
Accumulation product (SRRI = 0.15)

Ann. Returns	4.56%
Ann. Vol	18.0%
Sharpe ratio	0.25

Unfortunately, there was **no combination** of the initial indices that produced a suitable SRRI for the **high-risk** Income-type products.

QQ-Plot Analysis:

- Target portfolio returns deviate from normality.
- Exhibit heavier tails, skewness, and excess kurtosis.

Implications on Value at Risk:

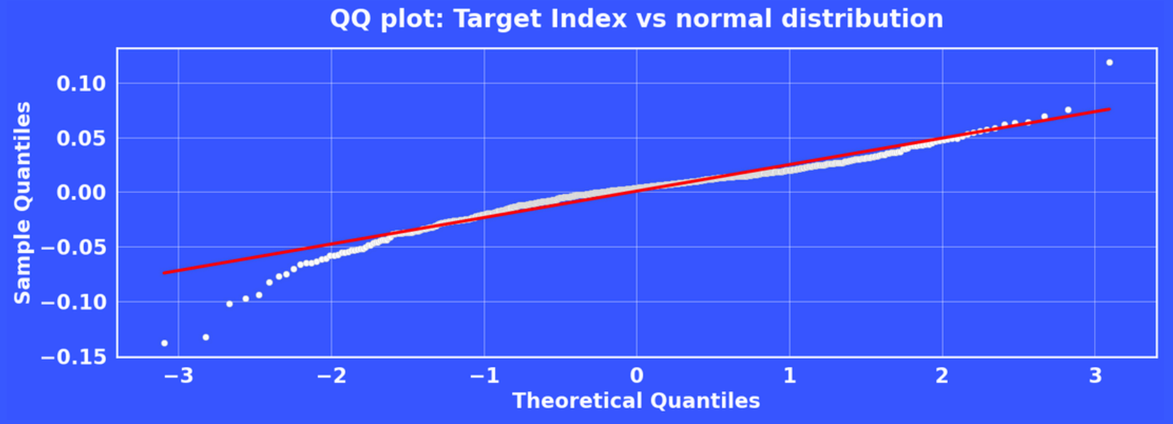
- the tail-sensitivity of VaR may lead to unreliable risk estimate

Solution:

- VaR estimated using the Cornish-Fisher expansion
  - Adjusts for skewness and kurtosis.
  - Reduces to Gaussian VaR if returns are normally distributed (i.e., skewness = 0, excess kurtosis = 0).

Regulatory Compliance:

- Adopted a conservative monthly VaR(99%) threshold of 15%, staying below 20% limit imposed by EU regulation.



Cornish-Fisher Adjusted Quantile (Z_{CF}):

$$Z_{\text{CF}} = Z_\alpha + \frac{1}{6}(Z_\alpha^2 - 1)S + \frac{1}{24}(Z_\alpha^3 - 3Z_\alpha)K - \frac{1}{36}(2Z_\alpha^3 - 5Z_\alpha)S^2$$

Cornish-Fisher Value at Risk (VaR_{CF}):

$$\text{VaR}_{\text{CF}} = -Z_{\text{CF}} \cdot \sigma - \mu$$

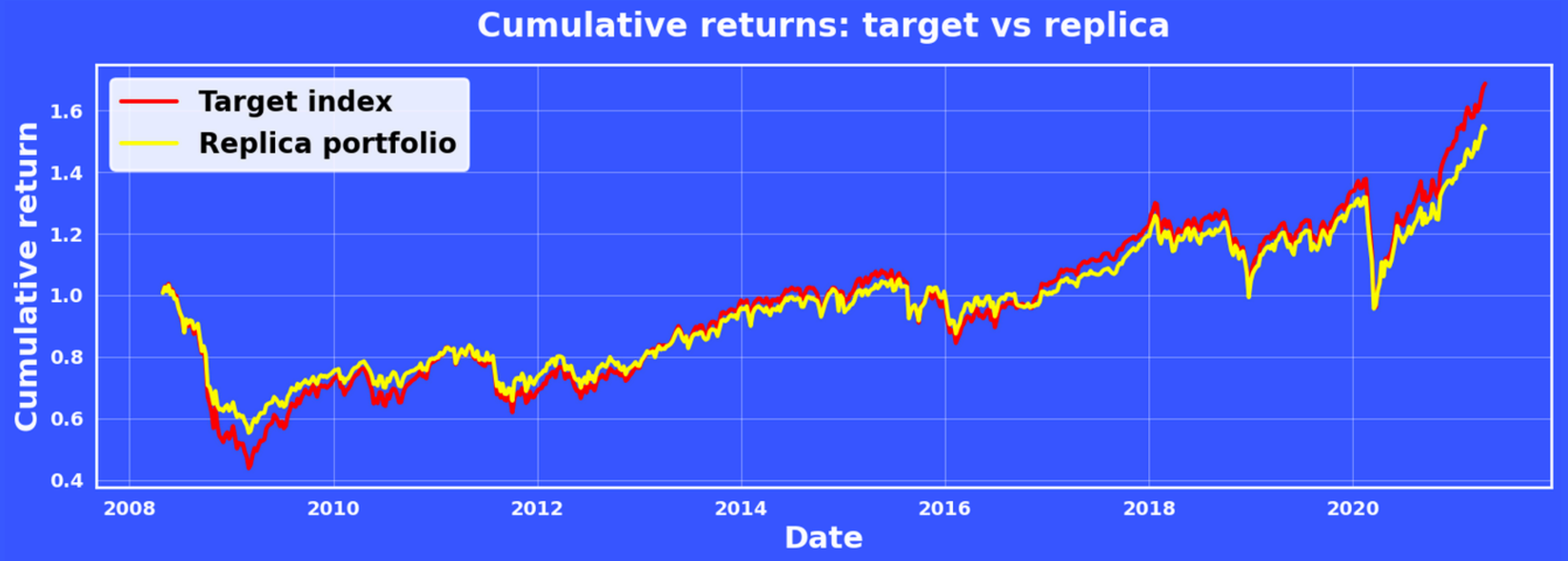
$$\mu = Returns\ Mean \cdot \Delta$$
$$\sigma : Returns\ Standard\ Deviation \cdot \sqrt{\Delta}$$
$$S = \frac{Returns\ Skewness}{\sqrt{\Delta}}$$
$$K = \frac{Returns\ Excess\ Kurtosis}{\Delta}$$
$$Z_\alpha = Standard\ Normal\ Quantile$$

We used **Linear Regression** without regularization to replicate the target portfolio. Moreover, we imposed a zero-bias term to ensure the portfolio remained self-financed. The OLS regression was run with different values for the rolling window length and the VaR computation window length. We selected the model that **minimised the difference between the target SRRI and the replica SRRI** in order to achieve our goal of obtaining a replica portfolio with the desired risk level.

	Metric	Target	Replica
0	Annualized return	5.64%	4.63%
1	Annualized volatility	17.87%	16.05%
2	SRRI	15.00%	14.99%
3	Sharpe ratio	0.32	0.29
4	Max Drawdown	57.80%	46.44%
5	Tracking Error	N/A	5.27%
6	Information ratio	N/A	-0.19
7	Average gross exposure	N/A	3.6776
8	Average VaR (1%, 1M)	N/A	11.35%

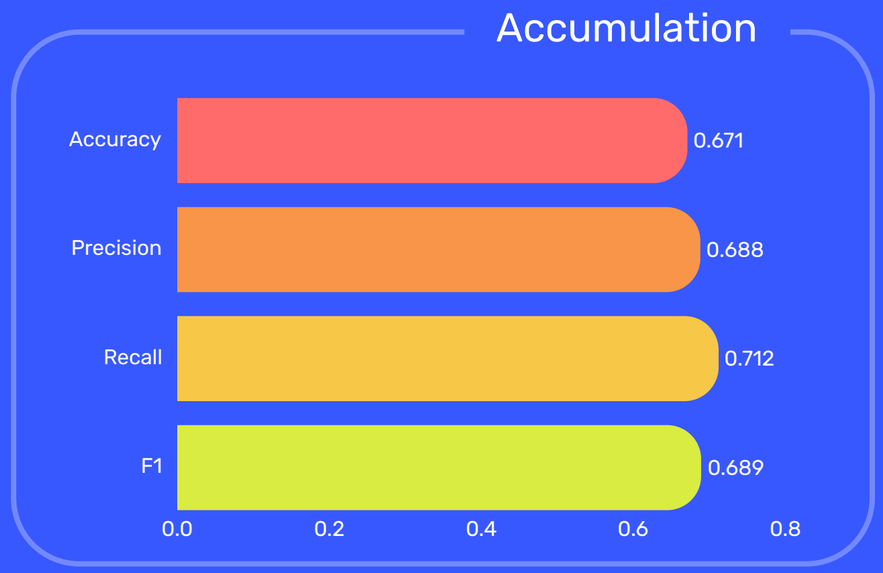
Best hyperparameters

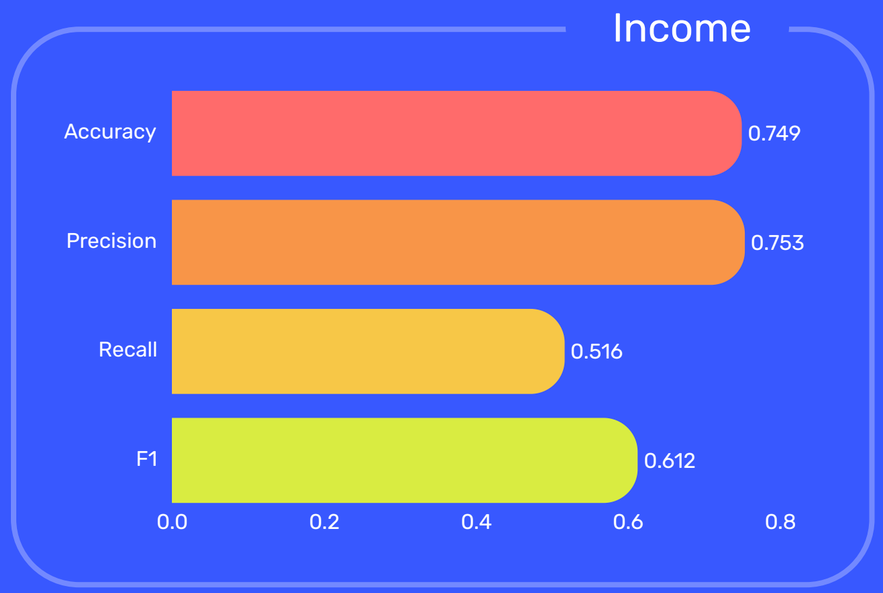
Rolling window	26
Recalibration window	124



ML

Baseline model

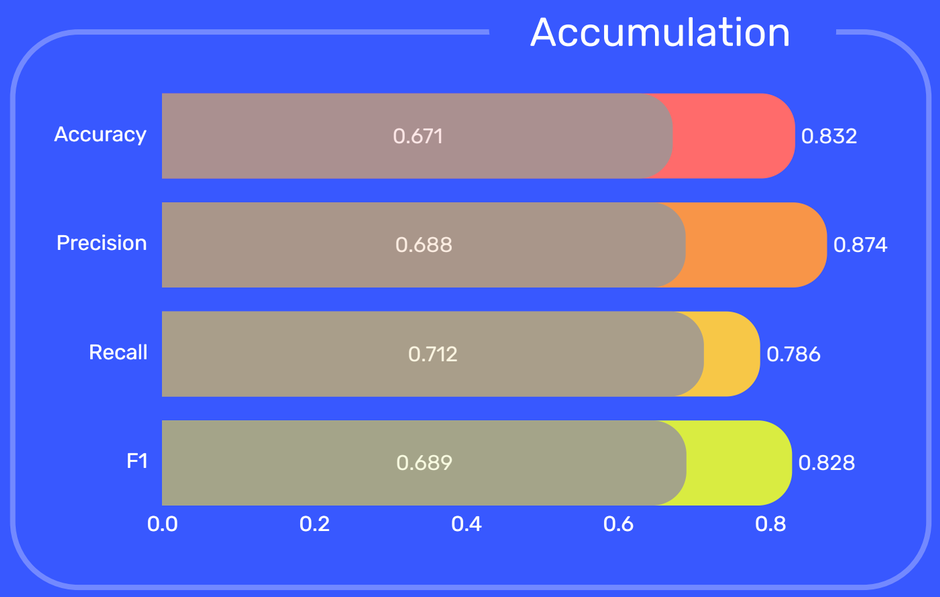




Logistic Regression

We used, as a baseline model for both targets, a Logistic Regression method, which provides a very good compromise between simplicity and effectiveness. Using the base features contained in the original dataset, except the Box-Cox transformed **Income** and **Wealth**, we get the metrics' values to be defeated. They are going to be plotted against every other model metric.

Accumulation's best: XGBoost

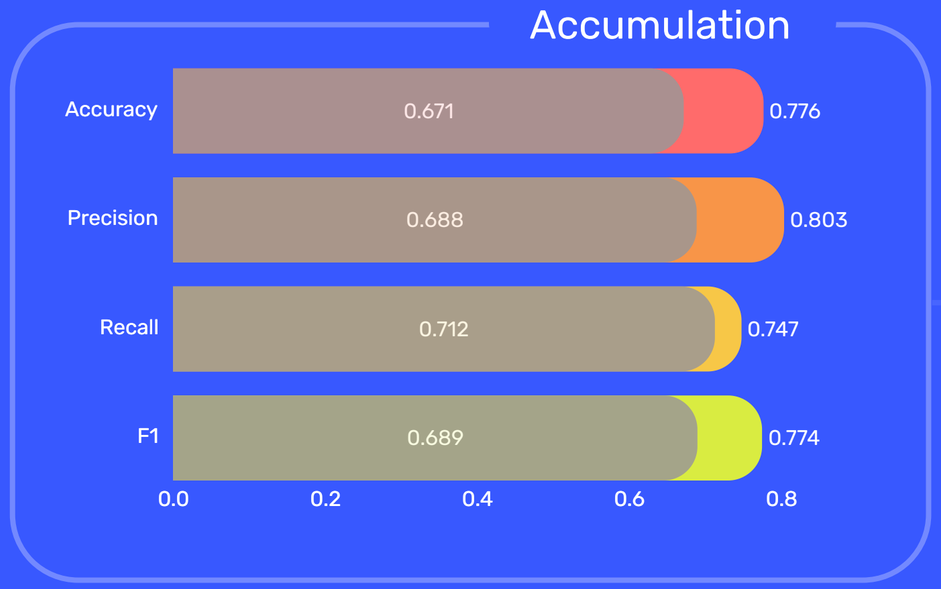


Our best model is **Extreme Gradient Boosting, XGBoost,** a particular implementation of gradient boosting known for accuracy and flexibility.

We selected the best possible set of features, including some engineered ones. The final model includes, other than base features, the **ratio** between **Wealth** and **Income**, the **product of log(RiskPropensity) and Wealth**, and the **ratio** between **FinancialEducation** and A**ge.**

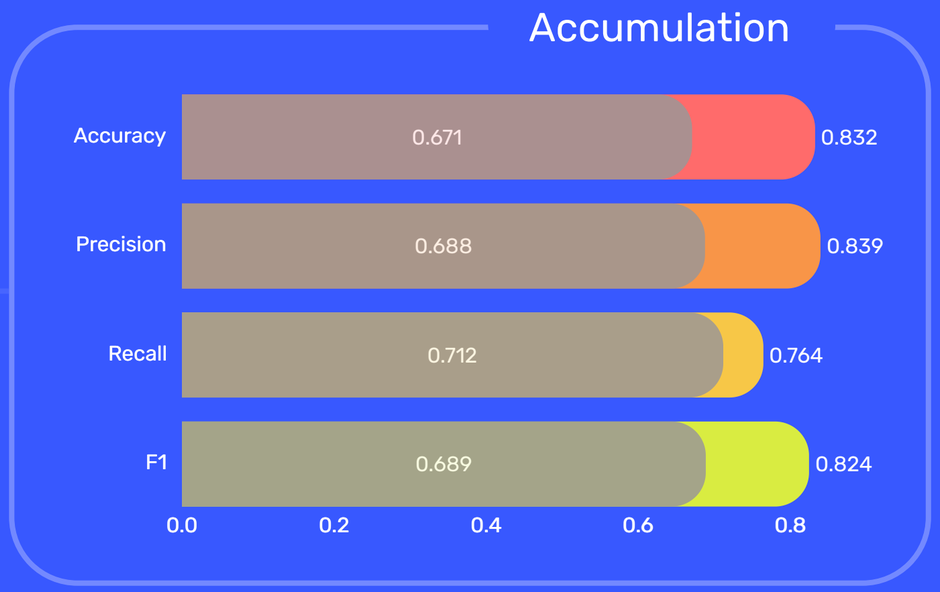
Further optimization included a gridsearch over model's hyperparameters.

More on Accumulation

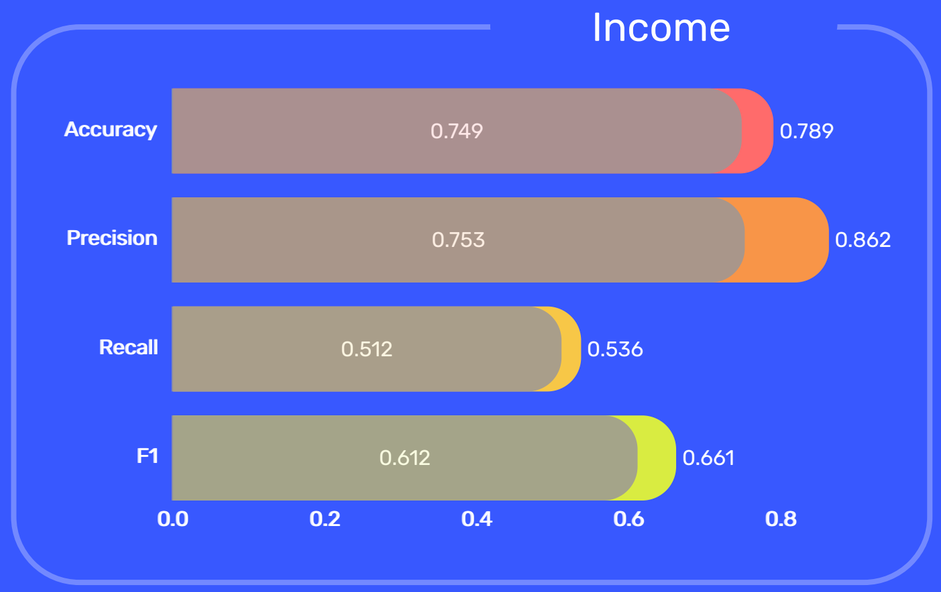


Another model that performed well is the **ANN.** There were no engineered features this time, only **Optuna** optimization, which led us to an architecture with 3 hidden layers and 97 hidden units.

Random forest had a very good performance, enhanced with a gridsearch over hyperparameters and the inclusion of **Income over Wealth ratio.**



Income’s best: XGBoost

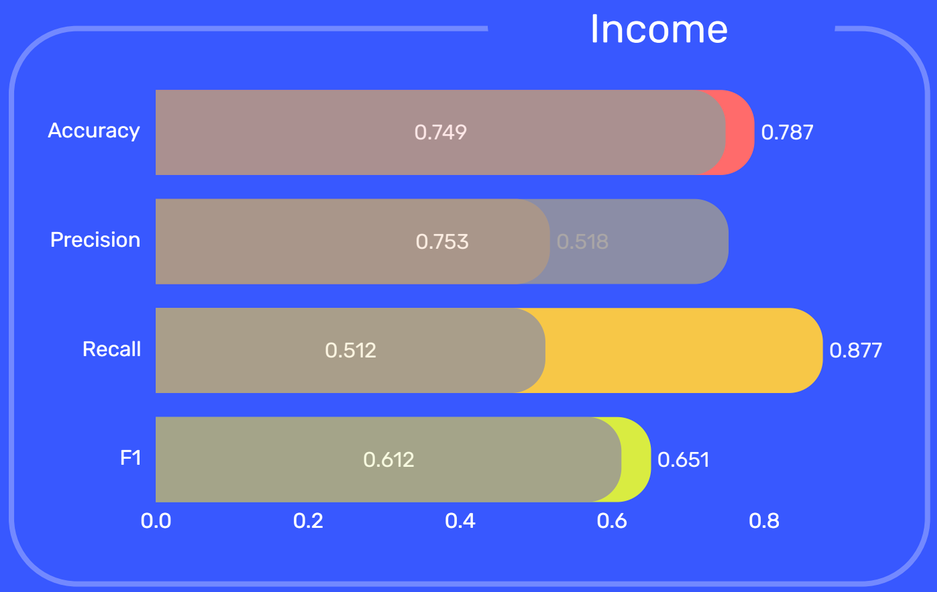


Our best model is yet again **Extreme Gradient Boosting.** Metrics do not improve as much with respect to the baseline as they did with Accumulation, as they suffer from class imbalance.

We selected the best possible set of features, including some engineered ones. The final model includes, other than base features, the **ratio** between **Wealth** and I**ncome.**

Further optimization included a gridsearch over model’s hyperparameters.

Model Ensemble for Income



We implemented a **voting system** among the best models. Other than the already mentioned XGBoost, we included a NN and a Random Forest.

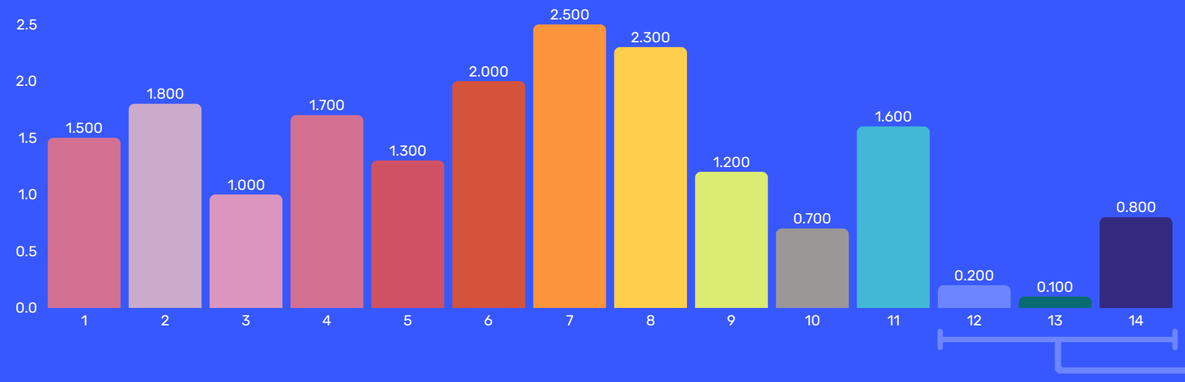
The model ensemble was implemented as a weighted voting mechanism based on **F1 score**.
We can see how some metrics improve, especially precision, but at the expense of recall that unperforms the baseline.

Which metric to optimize, then? We wanted not just numbers but something that could make us really understand the activity of the bank. As it didn't exist, we created one: **Business Impact Score.**

Business Impact Score

For a bank, clients are not all equal: wealthier clients impact the bank more.
The metric measures the **average per capita profit from product commissions**. We profit from correct recommendations (TP), we lose from missed gains (FN), and we are penalized for incorrect predictions (FP).

$$\frac{1}{N} \cdot \left(\sum_{TP} (comm \cdot wealth) - \alpha \sum_{FN} (\overline{comm} \cdot wealth) - \beta \sum_{FP} (\overline{comm} \cdot wealth) \right)$$



Which commissions did we use?
We did some industry research and associated some likely levels of commission for each type of product.

We also included our **newly created products**, the 2 types of time deposit and the replicated synthetic product.

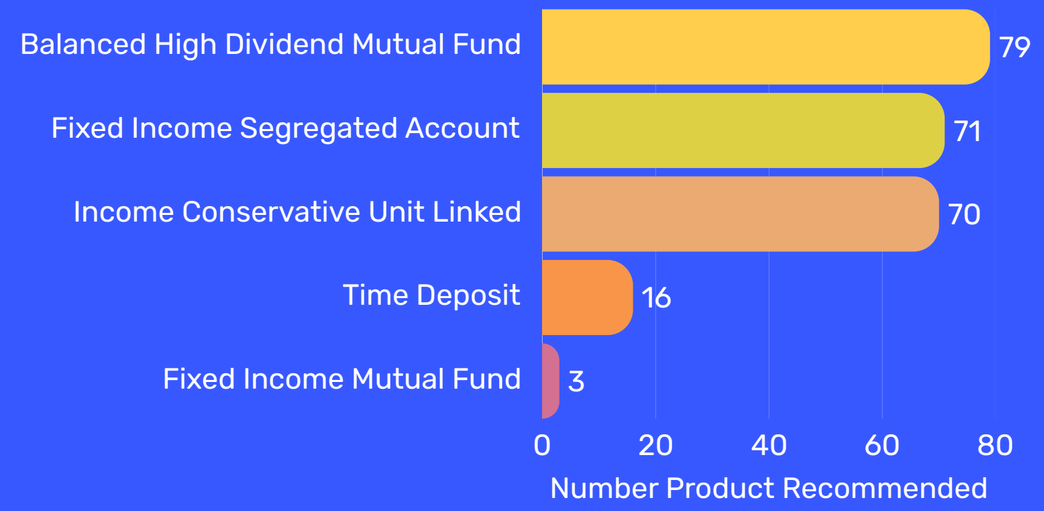
Income BIS	
Baseline	0.46k €
XGBoost	0.48k €
Ensemble	0.47k €

Accumulation BIS	
Baseline	0.41k €
XGBoost	0.57k €
Ensemble	0.45k €

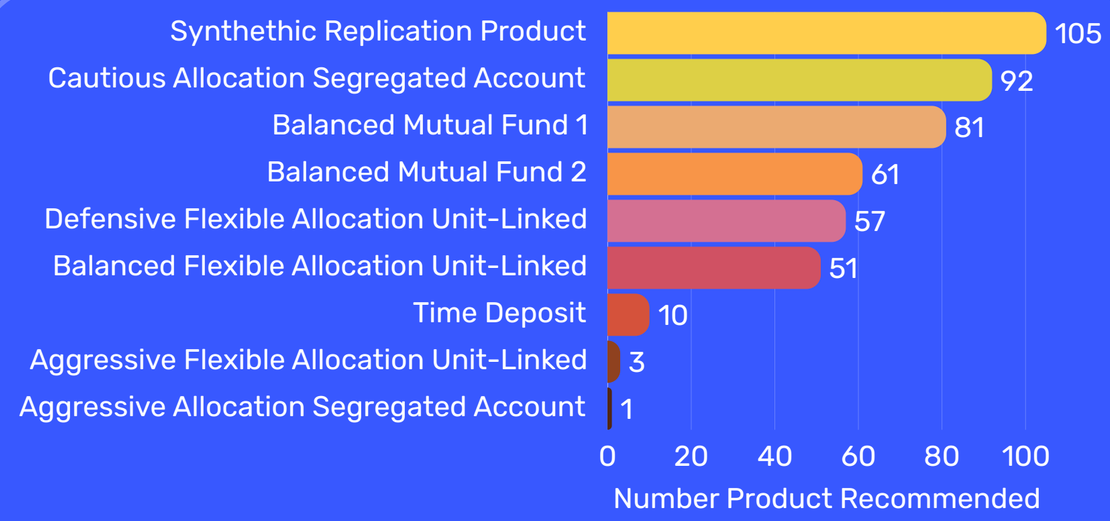
Recommender

The final step of our pipeline is product recommendation: for each client identified as willing to buy, we assign the first instrument with a risk level **below the client's propensity to risk.** The distribution of the recommended products allows the asset manager to increase the knowledge of its customers (KYC).

Income



Accumulation



The introduction of the time deposit enabled us to offer a product for **every client profile**. Moreover, the synthetic replication product for accumulation investment turned out to be the most frequently recommended, suggesting it may effectively capture a **real market segment**.

Conclusions

In conclusion, we were able to **assign a desired product to all clients** for both Income and Accumulation Investments. The newly created products were, in fact, able to **satisfy a considerable number of new clients**.

Thanks to the introduction of the **BIS**, we were able to create a **link between machine learning metrics and the real world** of a bank, made of profits and costs. BIS crowned the **XGBoost** as the **best possible model** for our classification problem for both Income and Accumulation.

A possible **new development** could be the **creation of a riskier product**, which, with our data, we were not able to replicate.