

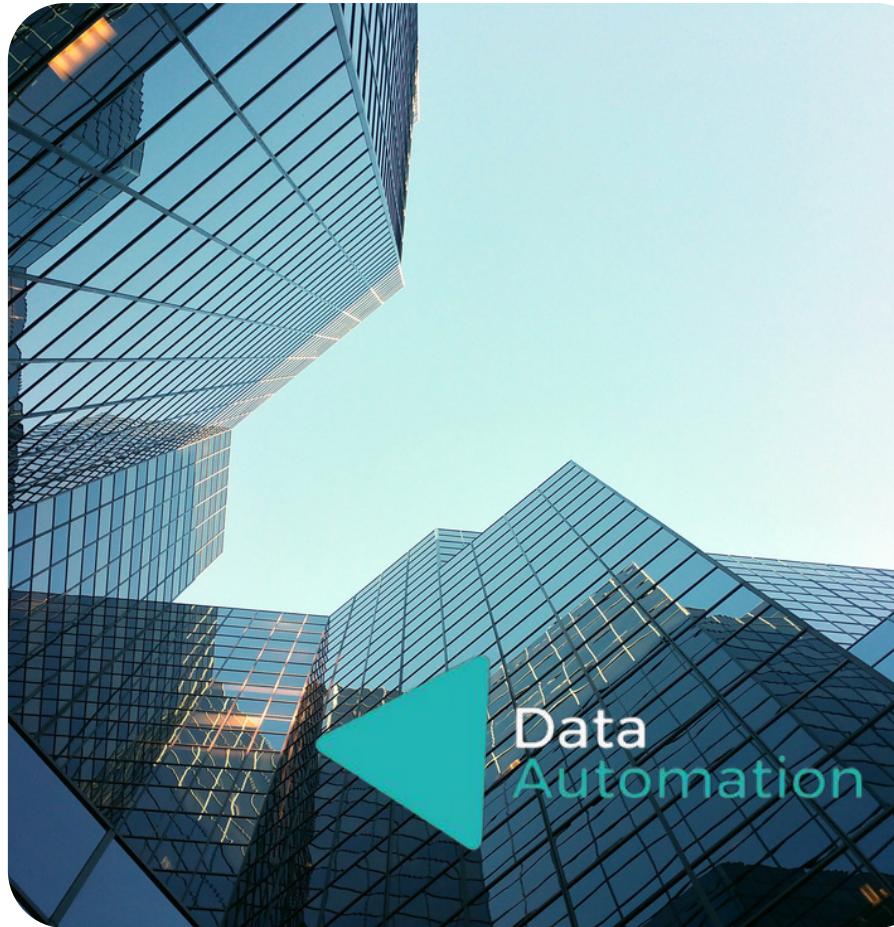
# Prediksi Pelanggan Bank yang Churn

Dautomation Consultant



# Kami dan Klien Kami

## Perkenalan



### Dautomation Consultant

Kami adalah tim analis dari Dautomation Consultant, sebuah perusahaan jasa profesional di bidang manajemen bisnis, termasuk didalamnya jasa terkait data science

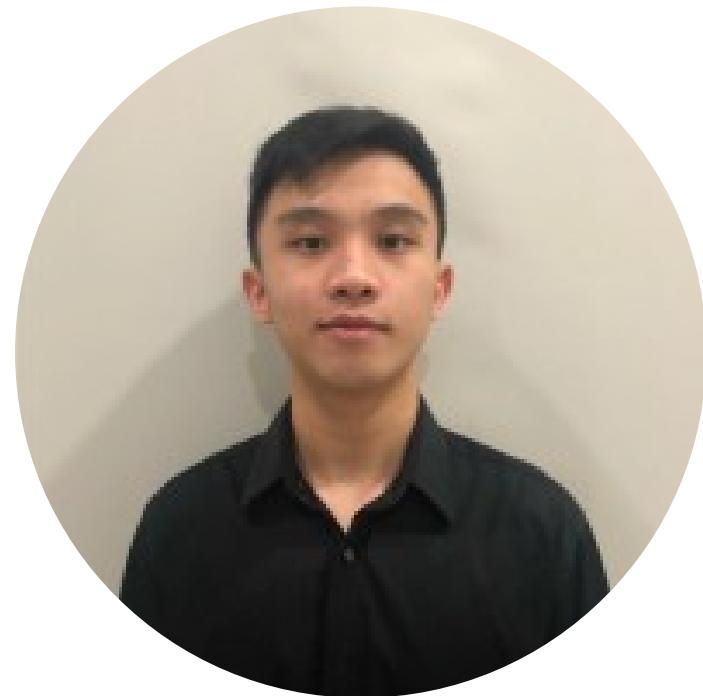


### European Money Bank

Sebuah Bank Eropa yang menyediakan jasa keuangan seperti tabungan, pinjaman, asuransi dan lain-lain



# Tim Kami



DARNELL KIKOO



PHILIP STANLEY



ILAN ASQOLANI



ZIRRASYI PUTRA



AJENG KINESTI

# Kerangka Laporan



- 01**  
Latar Belakang
- 02**  
Dataset (EDA & Pre-processing)
- 03**  
Ketidakpuasan terhadap Kartu Kredit
- 04**  
Ketidaksesuaikan dengan Ekspektasi
- 05**  
Modeling dan Evaluasi
- 06**  
Estimasi Impact + Rekomendasi Bisnis

# 1. Latar Belakang



**Apa yang terjadi?**

**20,37%** Customer churn dari European Money Bank

**Apa itu Customer Churn?**

Pelanggan yang meninggalkan suatu bisnis dalam waktu tertentu

**Apa yang akan terjadi jika customer churn ini dibiarkan?**

Penurunan profit/kerugian untuk jangka pendek dan mempengaruhi keberlanjutan bisnis bank

**Kenapa ini bisa terjadi?**

Kami mencurigai bahwa kualitas produk dan layanan perbankan tidak sesuai dengan ekspektasi customer

# Problem Statement

Terjadinya customer churn yang tinggi, yaitu mencapai 20,37% dari total customer

## Goal

Menurunkan jumlah customer churn

## Objective

Mendesain model Machine Learning supervised yang dapat memprediksi kemungkinan customer churn

## Business Metrics

Penurunan jumlah customer churn



# 2. Dataset (EDA)

RowNumber	
CustomerId	
Surname	
CreditScore	
Geography	
Gender	
Age	
Tenure	
Balance	
NumberOfProducts	
HasCrCard	
IsActiveMember	
EstimatedSalary	
Exited	✓

## Shape

14 kolom dan 10.000 baris. 'Exited' adalah Feature Target

## Data Exploration (Sampling)

Data untuk setiap feature konsisten.

## Distribusi Data

- Data di 4 feature relatif terdistribusi normal, sedangkan di 2 feature tidak
- Terdapat outliers

## Korelasi antar Feature

Tidak ditemukan adanya korelasi kuat antar feature

## Data Hilang (Missing Values) dan Duplikat

Tidak ditemukan data hilang dan duplikat

## Kesimpulan

Perlu menghilangkan outliers dan distribusi data di beberapa feature harus normal

## 2. Dataset (Pre-processing)

- **Drop unused columns**

Menghilangkan 3 Kolom yaitu Row Number, Customer ID, dan Surname karena tidak memiliki relevansi yang signifikan dalam proses analisis kami

- **Recast Data Types**

Mengubah 3 Kolom yaitu Geography, Gender, dan Credit Score menjadi Object dan Float.

Memasukan kedalam Fungsi Get Dummies agar dapat melihat keterangan dari nilai Geography dan Gender

- **Remove Outliers**

Menghapus Outliers untuk menghilangkan nilai yang ekstrem agar tidak mempengaruhi hasil analisis



## 2. Dataset (Pre-processing)



### Normalization

Menggunakan MinMaxScaler pada kolom Age, Tenure, Balance, Number of Products, Estimated Salary, dan Credit Score agar mendapatkan hasil yang adil

---

# Main Issue

## Persentase churn yang tinggi

European Money Bank memiliki persentase customer churn sebesar 20,37% dari total keseluruhan customer yang ada.

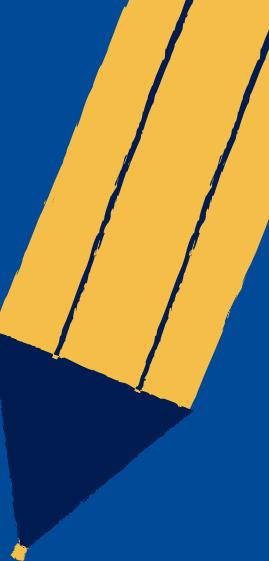
## Banyak terjadi gagal bayar pada produk credit card

Ditemukan sebanyak 69,9% customer yang churn memiliki credit card.

## Layanan dan kualitas produk yang tidak sesuai dengan ekspektasi customer

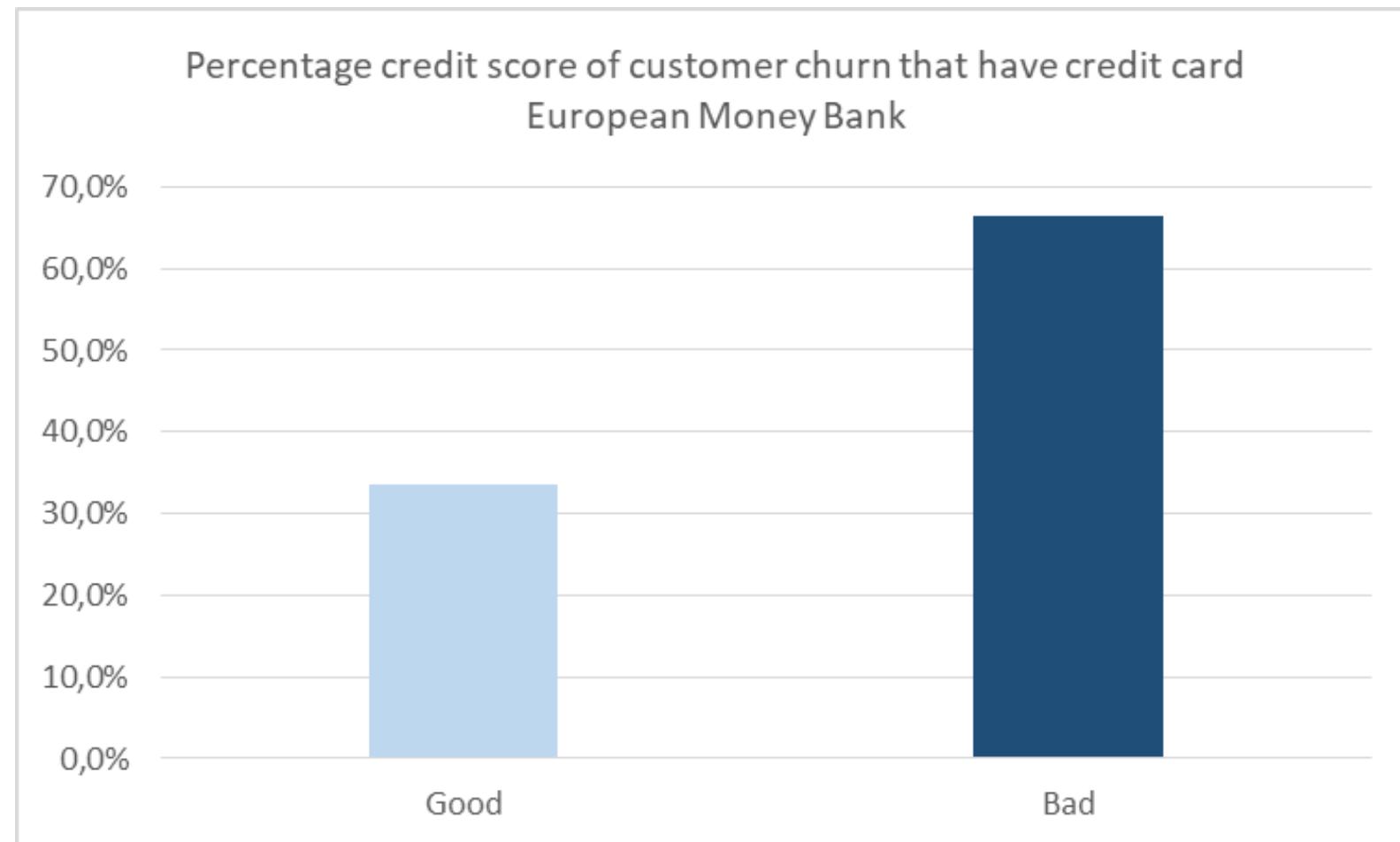
Hal ini dibuktikan dengan 69,2% customer yang churn, memiliki minimal satu produk perbankan.

**Dikhawatirkan ditahun-tahun kedepan, jika bank tidak mengevaluasi dan tidak melakukan perubahan, maka jumlah customer yang churn akan semakin tinggi. Dan ini tentu saja akan berimbas pada keberlanjutan bisnis bank**





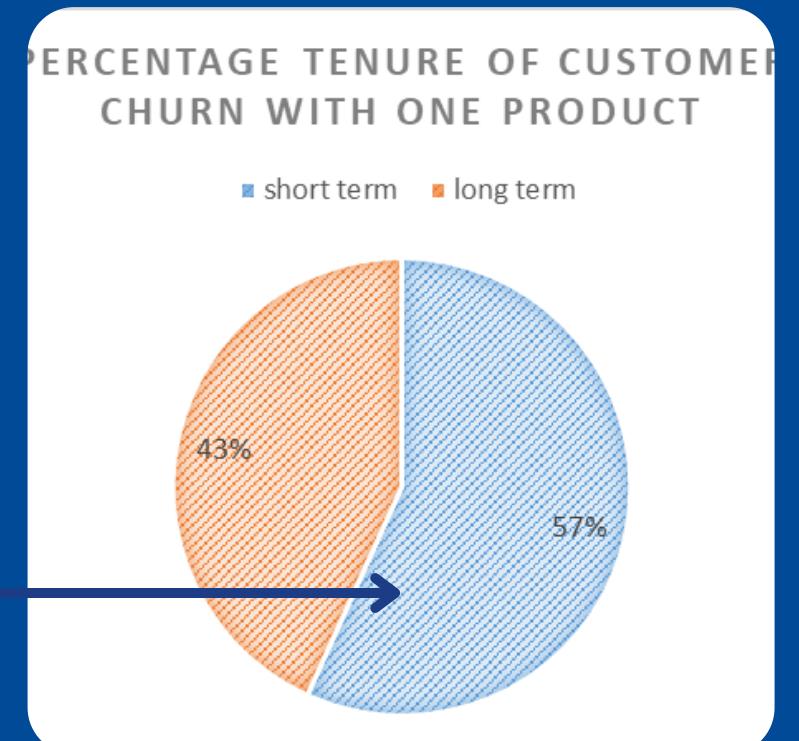
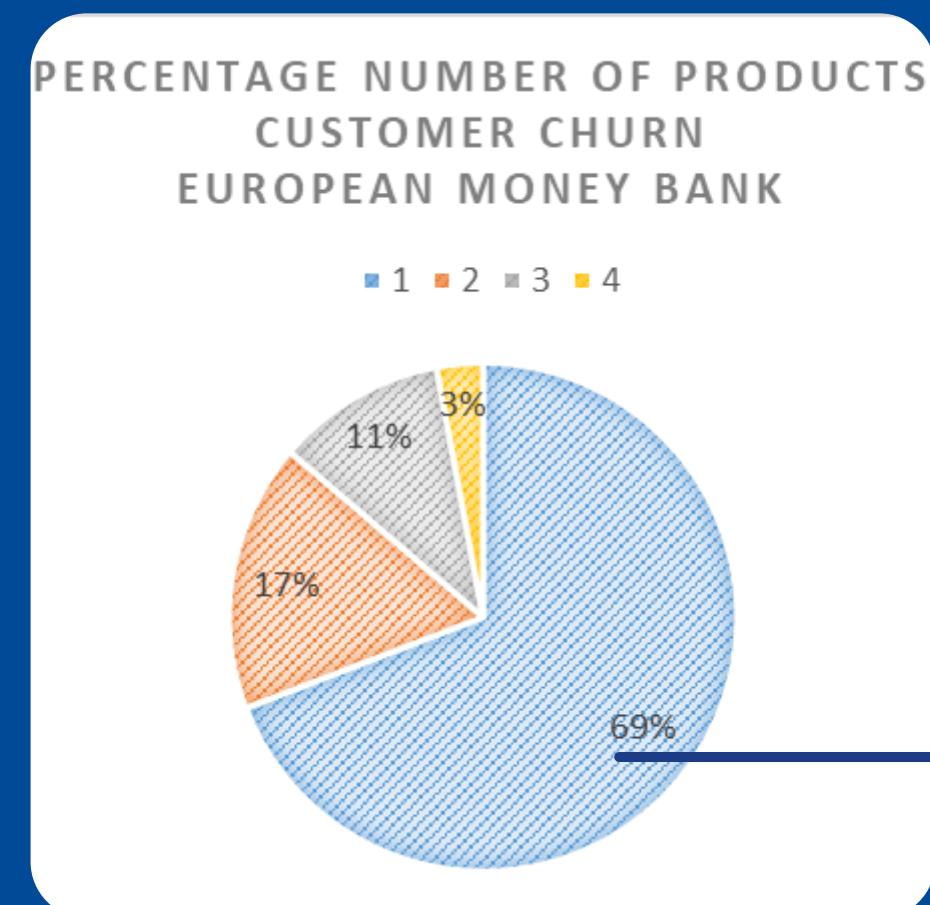
### 3. Kegagalan pembayaran pada credit card



Ditemukan sebanyak 69,9% customer yang churn memiliki credit card, dengan 66,4% customer memiliki credit score yang rendah/buruk. Untuk itu, kami mencurigai banyak terjadi gagal bayar pada produk credit card.

### 4. Ketidaksesuaian dengan ekspektasi

69,2% customer churn memiliki minimal 1 produk perbankan, dan 57%nya memiliki tenure jangka pendek (1-5 thn). Dengan demikian, dapat kami curigai bahwa terjadi ketidaksesuaian produk perbankan oleh customer sehingga tenure customer hanya bertahan dalam jangka pendek.



# Machine Learning: Before vs After

## Sebelum:

Bank mencoba menurunkan Churn Rate dengan cara melakukan Promosi dan Marketing ke **semua customer**

## Sesudah:

Bank dapat melakukan Promosi dan Marketing, ke **customer yang sudah diprediksi akan churn**, sehingga bisa **memperkecil biaya pengeluaran untuk tujuan Marketing**



# 5. Modelling dan Evaluasi



# Main Metrics: Recall

# Supporting Metrics : Precision, ROC, Accuracy, F1

## Alasan

Ingin menghindari **False Negative (FN)**, karena menghindari Customer yang dicap **tidak Churn** padahal seharusnya **Churn**

## Rencana Tindak Lanjut

Orang yang diprediksi **Churn**, akan difokuskan untuk peningkatan **CX**, ataupun **Ads** dan juga **Marketing**

		Real Label	
		Positive	Negative
Predicted Label	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Precision =  $\frac{\sum \text{TP}}{\sum \text{TP} + \text{FP}}$

Recall =  $\frac{\sum \text{TP}}{\sum \text{TP} + \text{FN}}$

Accuracy =  $\frac{\sum \text{TP} + \text{TN}}{\sum \text{TP} + \text{FP} + \text{FN} + \text{TN}}$



# Approach 1

# Traditional Training dan Testing

Menggunakan uji coba berbagai algoritma machine learning terhadap dataset training (cross validation) dan juga testing

# Membandingkan Beberapa Algoritma Machine Learning



## Menggunakan Cross Validation

Untuk mendapatkan algoritma machine learning terbaik, dilakukan perbandingan model terhadap **dataset training**, ditambah dengan metode **cross\_validation (CV = 5)** untuk membuat model menjadi lebih **robust**

# Hasil Cross Validation dan terhadap Testing Dataset

	Model	Recall	Precision	ROC_AUC	F1	Accuracy
0	Decision Tree	0.500346	0.470462	0.682512	0.484730	0.794124
1	LGBM	0.458466	0.703398	0.851563	0.554990	0.857522
2	CatBoost	0.452297	0.719691	0.856423	0.555388	0.859782
3	Adaboost	0.428951	0.702448	0.837661	0.532403	0.854200
4	Gradient Boosting	0.427569	0.738822	0.854763	0.541552	0.859915
5	XGBoost	0.421395	0.743975	0.854658	0.537843	0.859916
6	Random Forest	0.415906	0.730975	0.845087	0.530121	0.857257
7	Gaussian	0.385706	0.557908	0.798912	0.455878	0.821636
8	KNN	0.368566	0.568581	0.735882	0.447175	0.823499
9	LDA	0.308153	0.632193	0.790986	0.413724	0.830941
10	SVC	0.293774	0.777026	0.812624	0.426041	0.846890
11	Logistic Regression	0.275912	0.668344	0.790950	0.389801	0.832935
12	SGD	0.208144	0.704276	0.786621	0.288630	0.821372

## CV Terhadap Training

Dapat dilihat bahwa **Model-model** yang telah **diuji coba** masih memberikan hasil yang tidak optimal, dengan rata-rata **recall dibawah 0.5**

	Model	Recall	Precision	ROC_AUC	F1	Accuracy
0	Decision Tree	0.469780	0.447644	0.665345	0.458445	0.785221
1	CatBoost	0.456044	0.768519	0.711542	0.572414	0.868155
2	LGBM	0.436813	0.750000	0.700938	0.552083	0.862839
3	XGBoost	0.425824	0.786802	0.699069	0.552585	0.866560
4	Random Forest	0.423077	0.754902	0.695059	0.542254	0.861776
5	Gradient Boosting	0.409341	0.788360	0.691486	0.538879	0.864434
6	Adaboost	0.401099	0.741117	0.683740	0.520499	0.856991
7	Gaussian	0.395604	0.580645	0.663524	0.470588	0.827751
8	KNN	0.335165	0.523605	0.630997	0.408710	0.812334
9	SVC	0.299451	0.819549	0.641815	0.438632	0.851675
10	LDA	0.293956	0.690323	0.631157	0.412331	0.837852
11	Logistic Regression	0.271978	0.697183	0.621816	0.391304	0.836257
12	SGD	0.145604	0.815385	0.568847	0.247086	0.828283

## Terhadap Testing



# Approach 2

## Add Weighted Parameter

Menambahkan parameter yang memberikan weight lebih kepada class minoritas (biasanya kelas positif). Contohnya `scale\_pos\_weight` dan `class\_weight`

## **``scale\_post\_weight``**

Didapatkan dengan cara membagi  
**jumlah class negative / class positif**

## **``class\_weight``**

Didapatkan dengan cara memberikan weight secara manual menggunakan dictionary, dengan value yang sama terhadap class **positif seperti di ``scale\_pos\_weight``**

```
scale_pos_weight = y_train.value_counts()[0]/y_train.value_counts()[1]
class_weight = {0: 1, 1: y_train.value_counts()[0]/y_train.value_counts()[1]}
```

# Hasil Model menggunakan Weighted Parameter



	Model	Recall	Precision	ROC_AUC	F1	Accuracy
0	XGBoost	0.739201	0.478102	0.854948	0.580443	0.793063
1	SGD	0.726089	0.379385	0.783339	0.493712	0.711319
2	SVC	0.715850	0.462457	0.836147	0.561761	0.783892
3	Logistic Regression	0.711714	0.390575	0.791244	0.504198	0.728866
4	LGBM	0.689764	0.520654	0.849728	0.593286	0.816987
5	CatBoost	0.689091	0.514766	0.853888	0.589131	0.814062
6	Decision Tree	0.469472	0.462238	0.669217	0.465787	0.791602
7	Adaboost	0.428951	0.702448	0.837661	0.532403	0.854200
8	Gradient Boosting	0.427569	0.738822	0.854763	0.541552	0.859915
9	Random Forest	0.396691	0.747297	0.844518	0.518144	0.857124
10	Gaussian	0.385706	0.557908	0.798912	0.455878	0.821636
11	KNN	0.368566	0.568581	0.735882	0.447175	0.823499
12	LDA	0.308153	0.632193	0.790986	0.413724	0.830941

	Model	Recall	Precision	ROC_AUC	F1	Accuracy
0	SVC	0.730769	0.457045	0.761232	0.562368	0.779904
1	XGBoost	0.730769	0.468310	0.765846	0.570815	0.787347
2	LGBM	0.692308	0.500000	0.763095	0.580645	0.806486
3	CatBoost	0.686813	0.515464	0.765951	0.588928	0.814460
4	Logistic Regression	0.675824	0.384977	0.708380	0.490528	0.728336
5	SGD	0.662088	0.378336	0.700523	0.481518	0.724083
6	Decision Tree	0.431319	0.481595	0.659957	0.455072	0.800106
7	Gradient Boosting	0.409341	0.788360	0.691486	0.538879	0.864434
8	Adaboost	0.401099	0.741117	0.683740	0.520499	0.856991
9	Gaussian	0.395604	0.580645	0.663524	0.470588	0.827751
10	Random Forest	0.379121	0.750000	0.674399	0.503650	0.855396
11	KNN	0.335165	0.523605	0.630997	0.408710	0.812334
12	LDA	0.293956	0.690323	0.631157	0.412331	0.837852

CV Terhadap Training

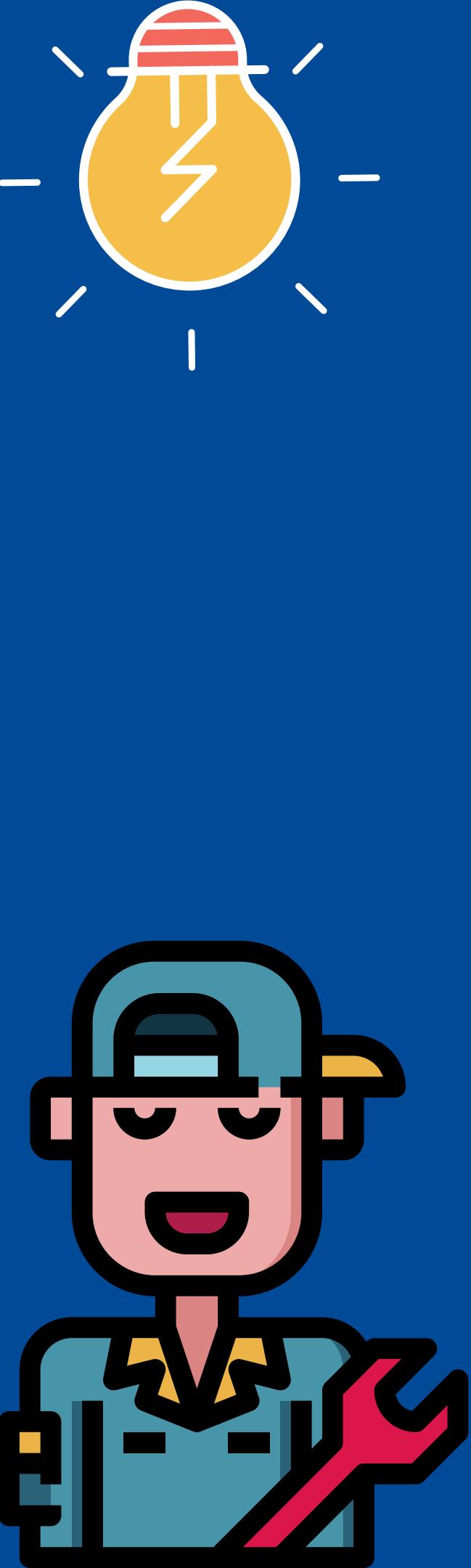
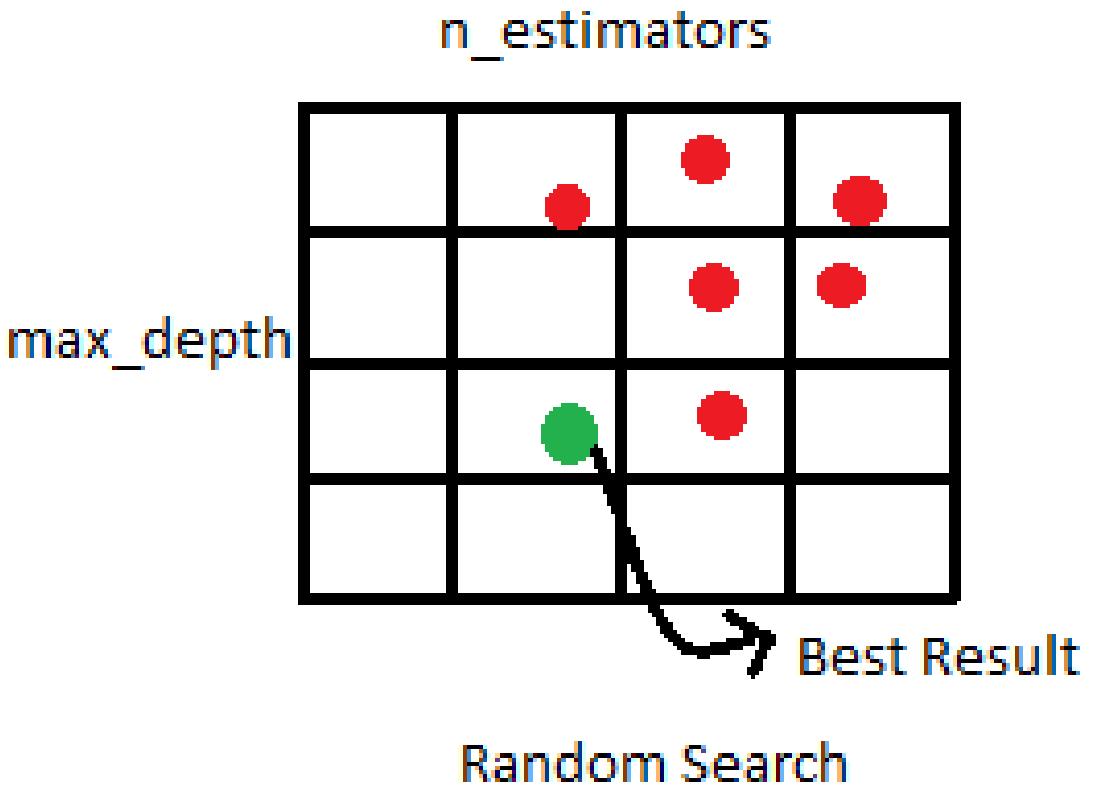
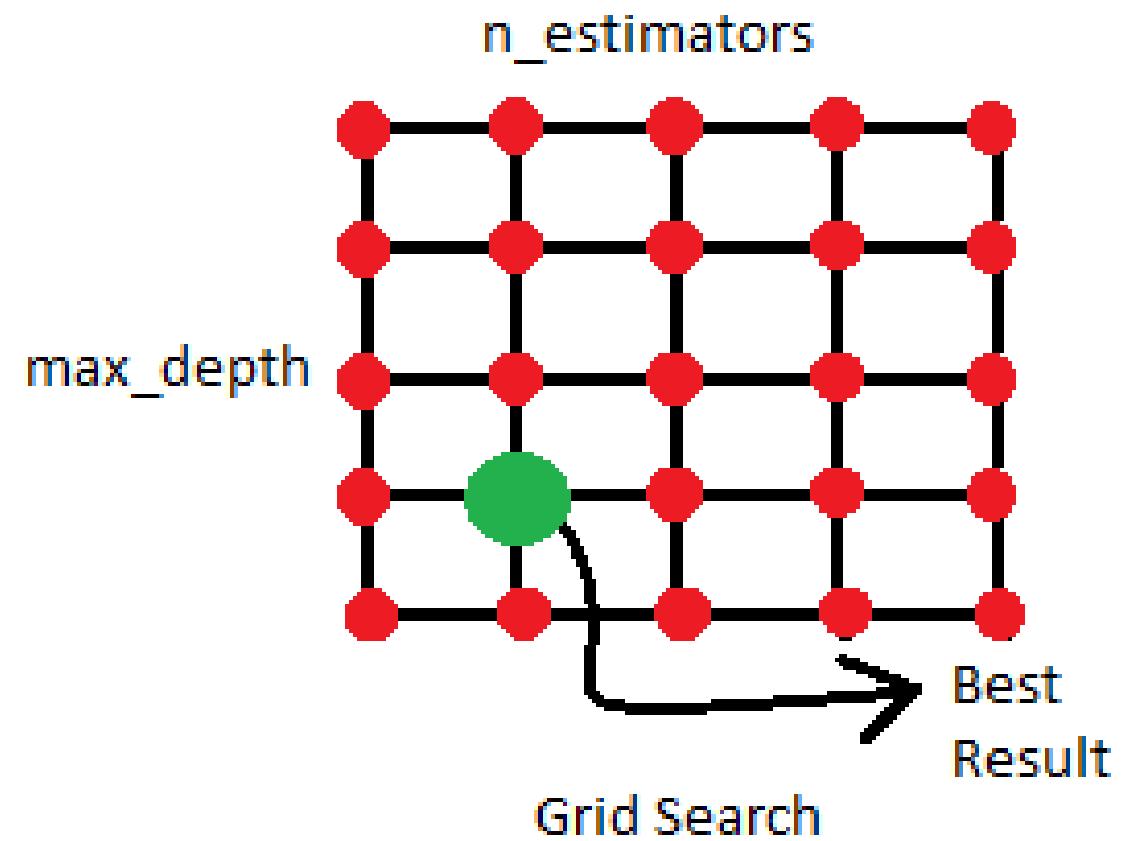
Terhadap Testing

Dapat dilihat setelah melakukan weight tuning, model **XGBoost** cukup mendominasi di **Recall dan juga supported metrics** lainnya. Maka dari itu akan digunakan untuk **hyperparameter tuning** lebih lanjut

# Hyperparameter Tuning

- Dilakukan untuk menentukan hyperparameter yang optimal.
- Setiap model memiliki hyperparameter yang berbeda-beda.
- Hyperparameter tuning dapat dilakukan dengan 2 cara, yaitu

Grid Search dan Randomized Search.



# Grid Search



Hyperparameters yang dipakai dalam projek ini:

- max\_depth
- n\_estimators

- colsample\_bytree
- colsample\_bylevel

- booster
- learning rate

- reg\_lambda
- reg\_alpha

Tidak dilakukan secara sekaligus untuk menghemat komputasi.  
Dilakukan secara bertahap (per 2 hyperparameters).

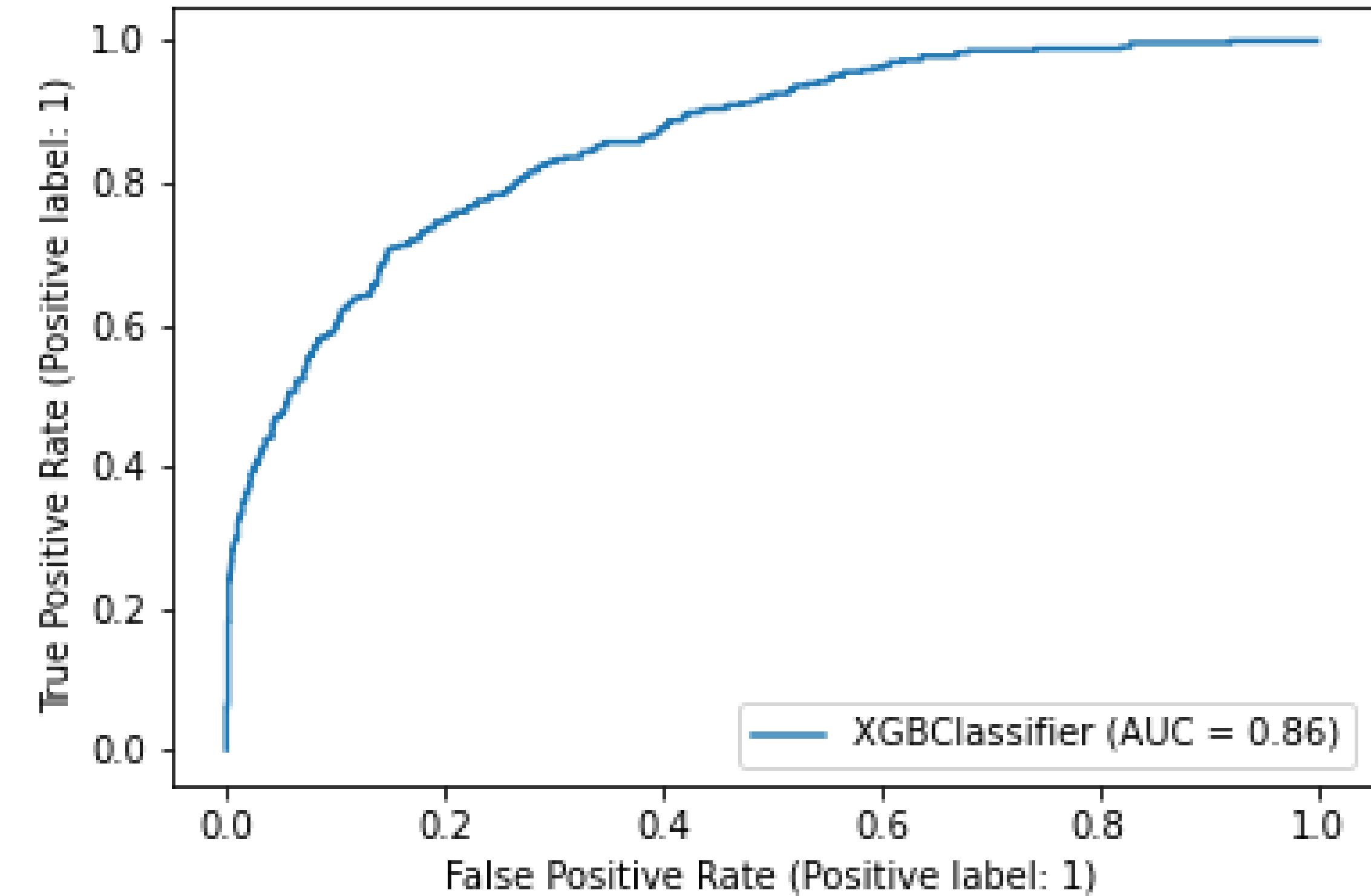
# Best Hyperparameters

1. **max\_depth = 2**
2. **n\_estimators = 200**
3. **colsample\_bylevel = 0.6**
4. **colsample\_bytree = 0.4**
5. **booster = gbtree**
6. **learning\_rate = 0.1**
7. **reg\_lambda = 0.001**
8. **reg\_alpha = 0.1**

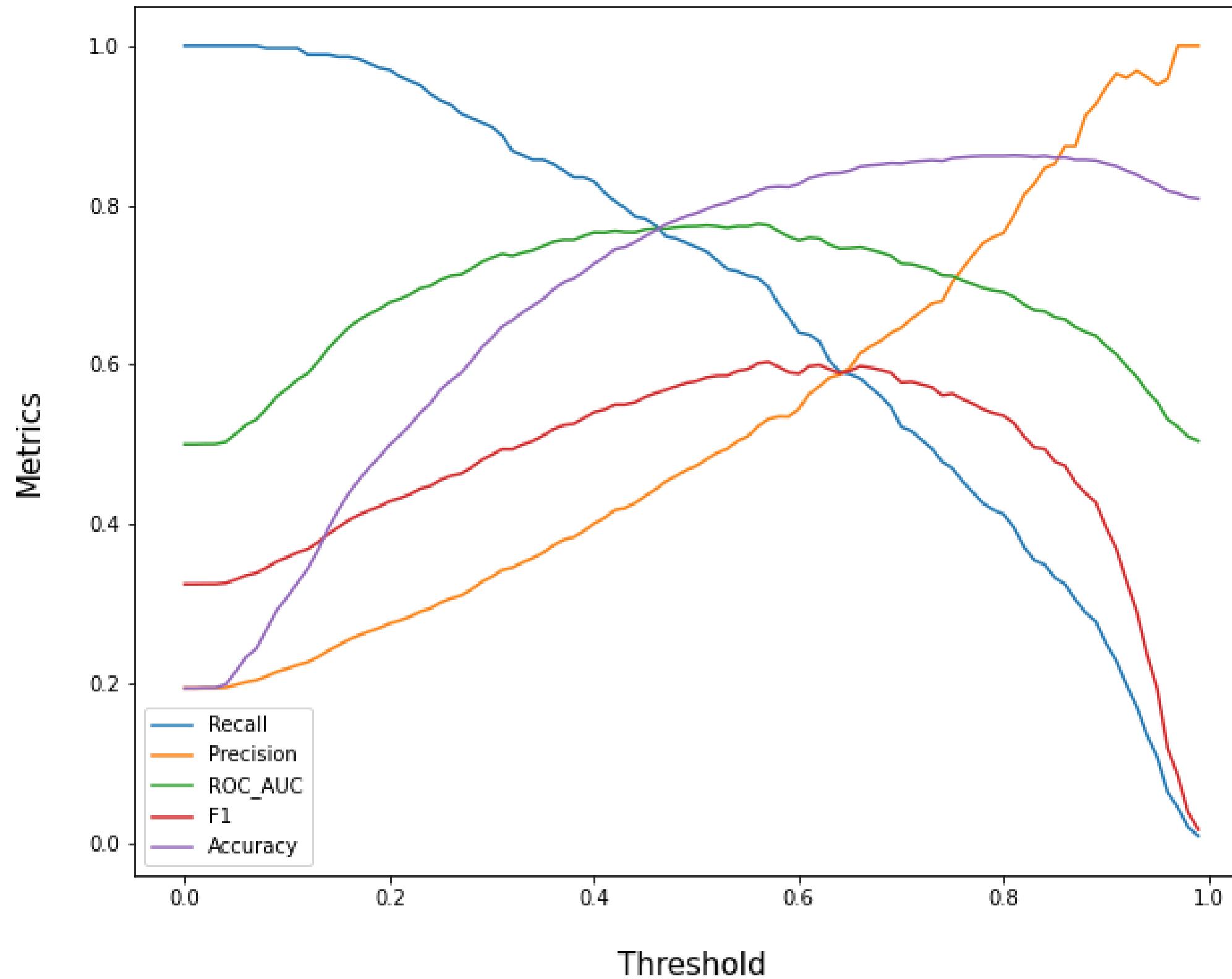
# Metrics

- **Recall: 0.73**
  - **Precision: 0.46**
  - **ROC AUC: 0.76**
  - **F1: 0.56**
  - **Accuracy: 0.78**
- 
- **Recall: 0.75**
  - **Precision: 0.47**
  - **ROC AUC: 0.77**
  - **F1: 0.58**
  - **Accuracy: 0.79**

# ROC Curve



# Metrics on Different Threshold



# Asumsi untuk analisis impact pada model ini:

---

## — **Marketing Efficiency = 50%** —

Persentase orang yang tidak jadi exit karena diberikan marketing, promo, ads, dan dicontact oleh customer service.

## — **Rate of Annoyance= 10%** —

Persentase orang yang awalnya tidak exit, tetapi menjadi exit karena marketing, promo, ads, dan contact customer service yang kurang tepat.

## — **Profit per Person = EUR 10** —

## — **Marketing Cost per Person = EUR 0.5** —



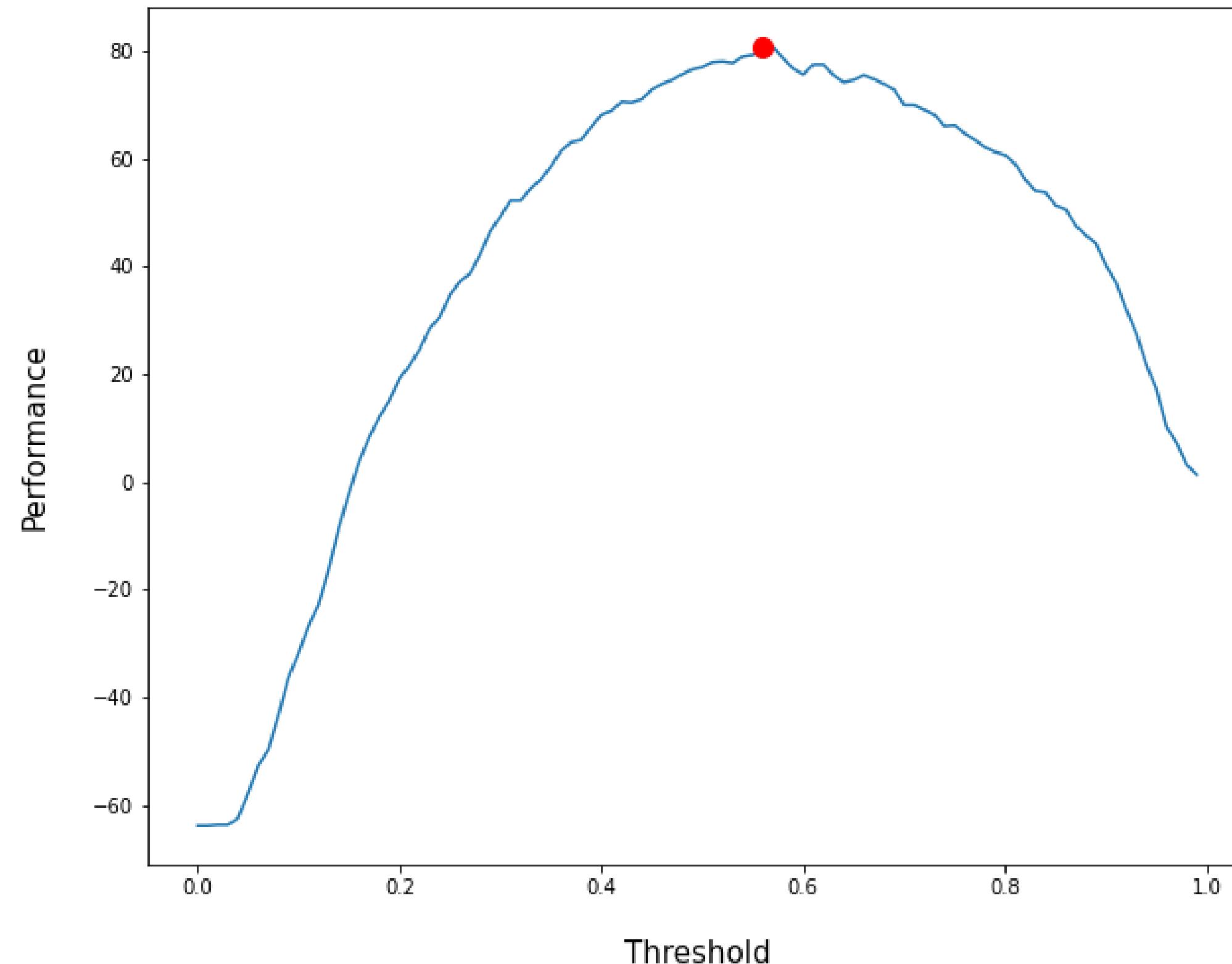


# Asumsi untuk analisis impact pada model ini:

**profit\_per\_person \* (true\_positive \* marketing\_eff) -**  
**profit\_per\_person \* (false\_positive \* ro\_annoyance) -**  
**marketing\_cost\_per\_person \* (true\_positive + false\_positive)**

**10 \* (true\_positive \* 0.5) -**  
**10 \* (false\_positive \* 0.1) -**  
**0.5 \* (true\_positive + false\_positive)**

# Model Impact on Different Threshold



# Pemilihan Threshold

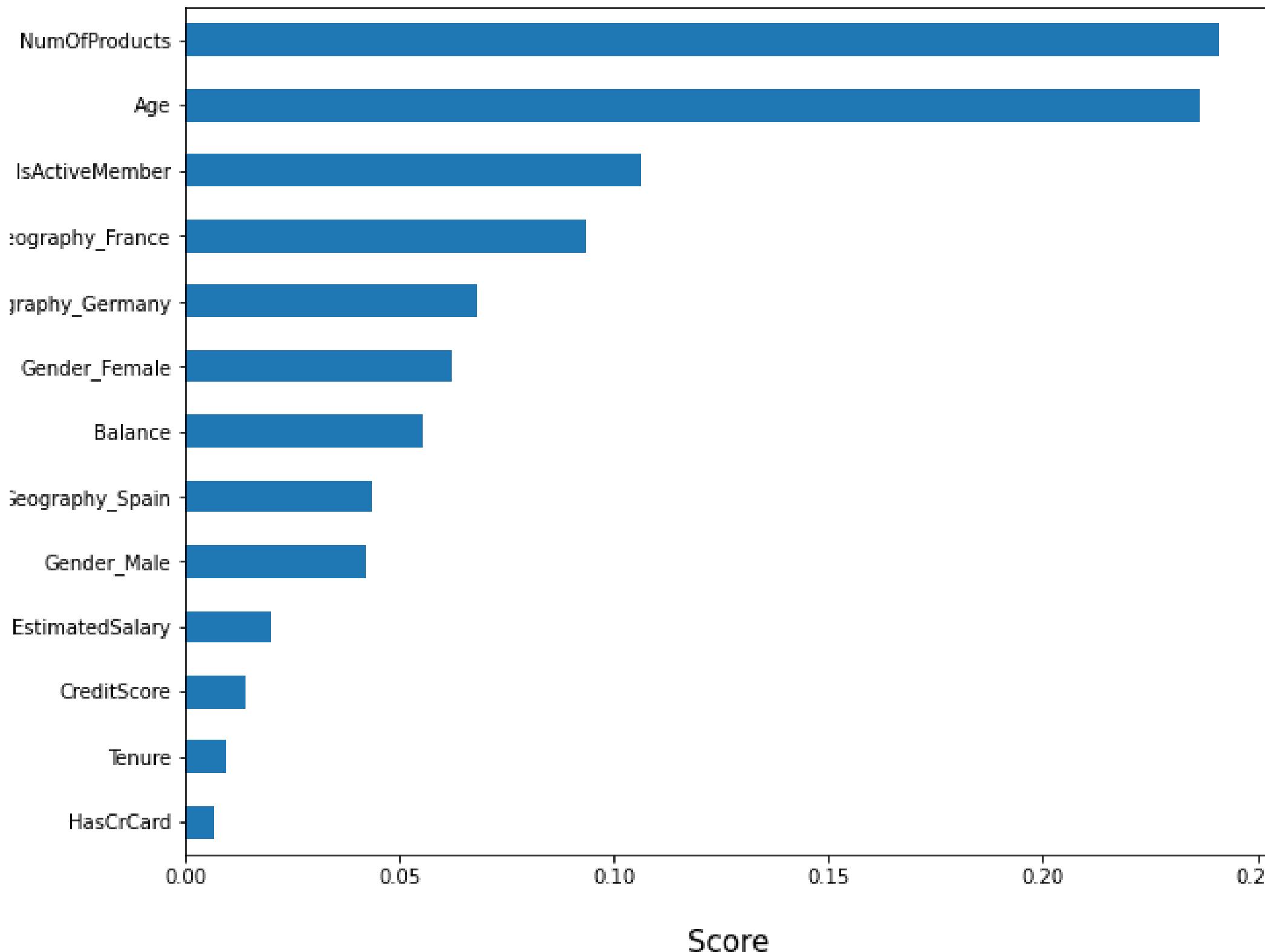
Kami memilih threshold dengan **impact tertinggi**.  
Impact paling tinggi saat threshold di level **0.56**

- Recall: 0.75
- Precision: 0.47
- ROC AUC: 0.77
- F1: 0.58
- Accuracy: 0.79

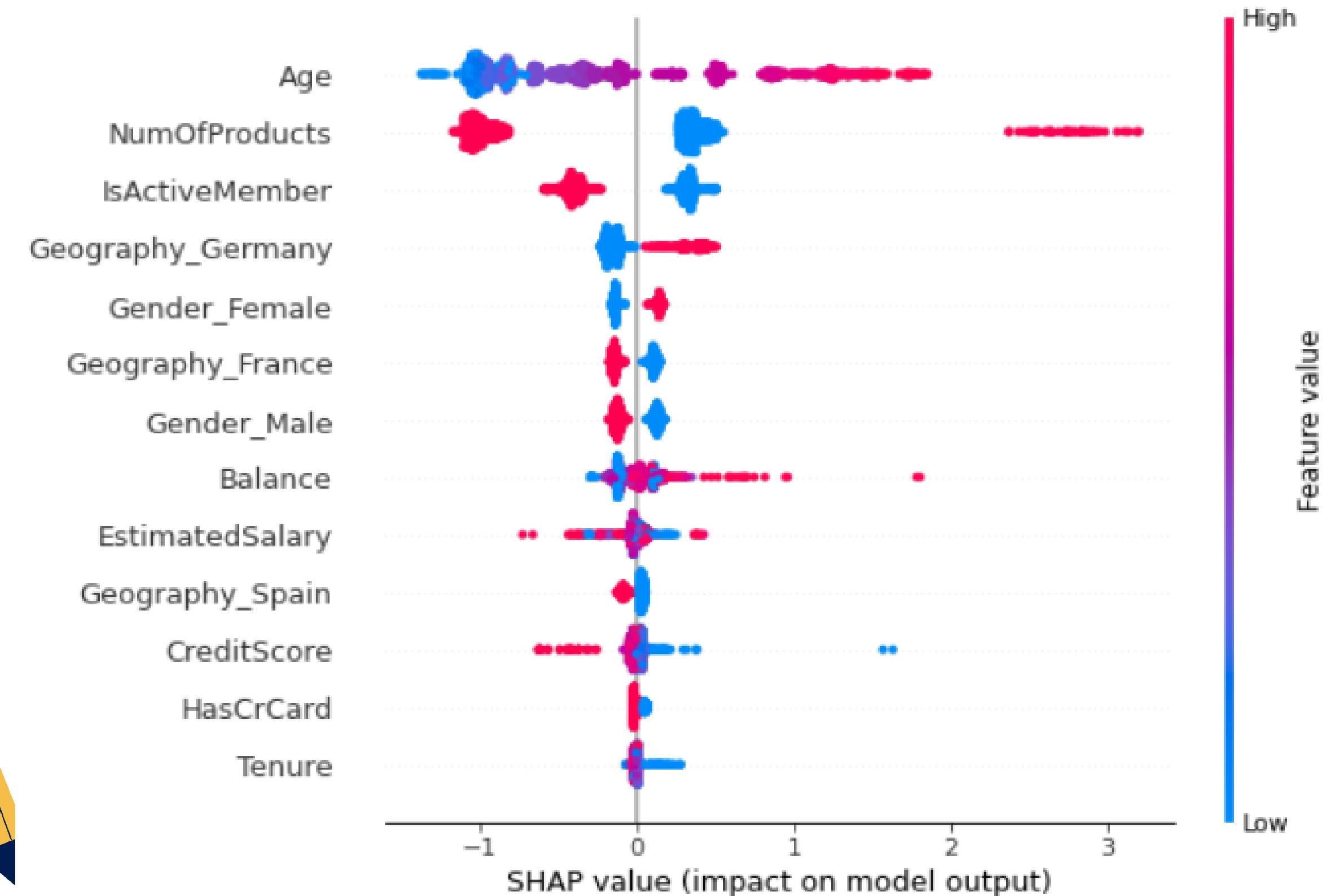
- Recall: 0.71
- Precision: 0.52
- ROC AUC: 0.78
- F1: 0.60
- Accuracy: 0.82



# Feature Importance



# SHAP Values



# – 6. Analisis dan Perbandingan Impact –

Simulasi pada Test Set (1881 customers)

## All-Exit Assumption

**True Positive = 364**

**False Positive = 1517**

$$10 * (364 * 0.5) - 10 * (1517 * 0.1) - 0.5 * (364 + 1517) = - \text{EUR } 637.5$$

## All-Stay Assumption

**True Positive = 0**

**False Positive = 0**

$$10 * (0 * 0.5) - 10 * (0 * 0.1) - 0.5 * (0 + 0) = \text{EUR } 0$$

## Dengan Model

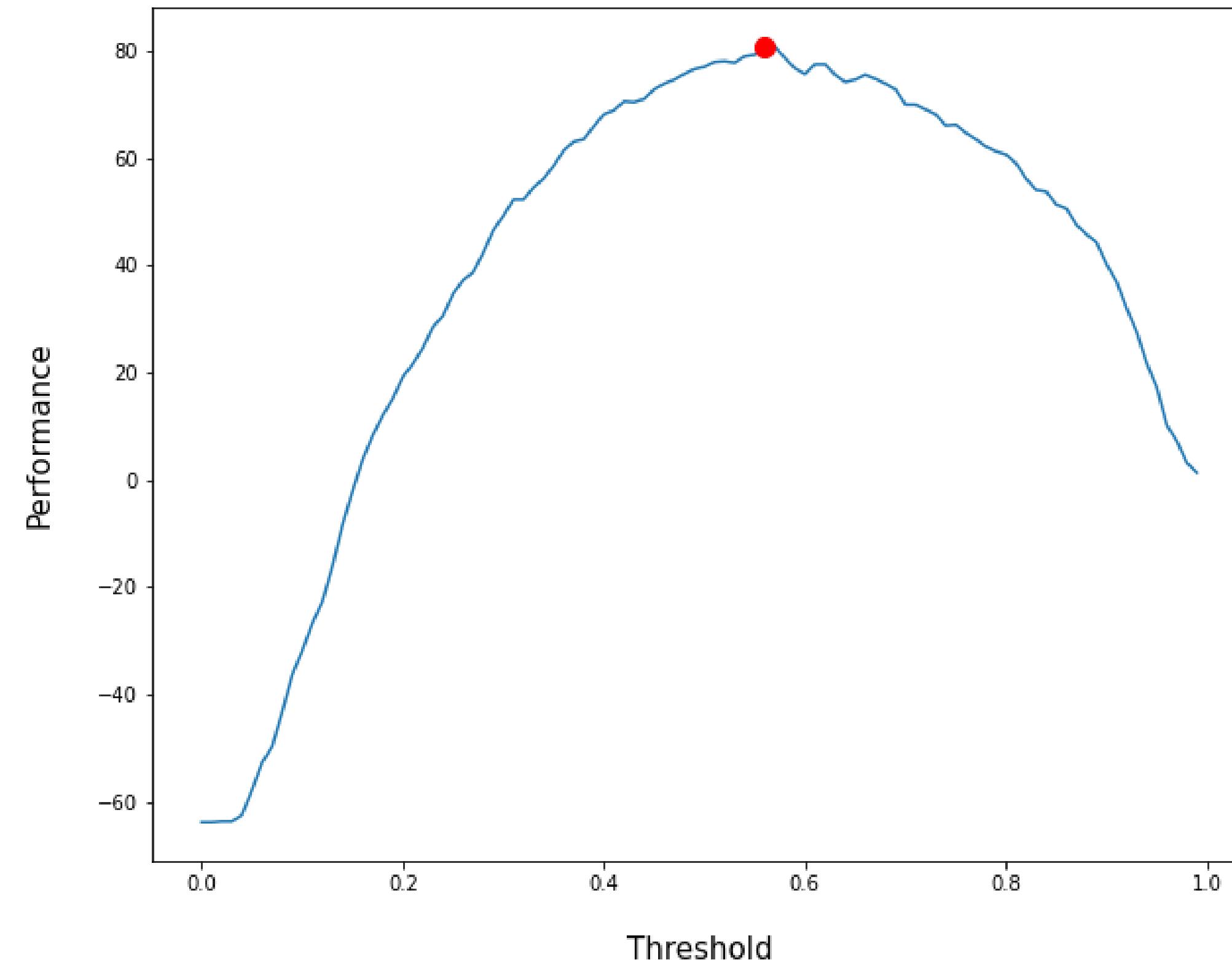
**True Positive = 258**

**False Positive = 236**

$$10 * (258 * 0.5) - 10 * (236 * 0.1) - 0.5 * (258 + 236) = \text{EUR } 807$$



# Model Impact on Different Threshold



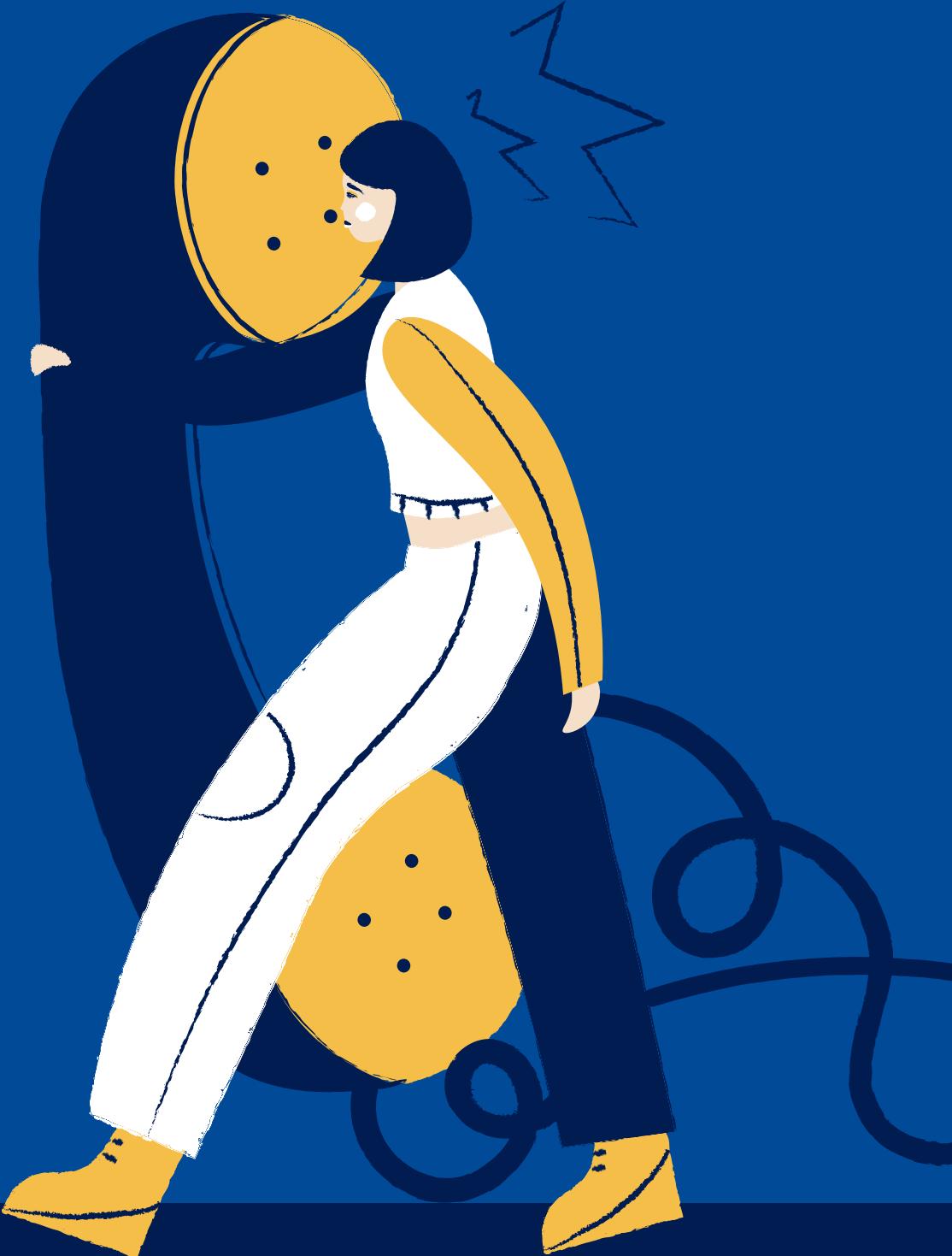
# 6. Rekomendasi Bisnis

## Tim Marketing

- Meningkatkan penawaran produk perbankan pada customer dengan usia yang produktif
- Meningkatkan penawaran produk perbankan lainnya seperti deposito, investasi, asuransi, dan lain-lain.

## Meningkatkan layanan & kualitas produk

- Membuat fast call service untuk penanganan masalah produk perbankan dengan cepat
- Meningkatkan mekanisme pembayaran online yang bekerja dengan baik untuk menghindari risiko gagal bayar pada produk credit card
- Memberikan diskon/reward untuk customer, sehingga dapat meningkatkan aktivitas dan tenure mereka.



# THANK YOU



# Referensi

<https://sci-hub.se/10.1007/978-981-15-5243-4> page:148-167

