Call-Center Campaign
Cost Optimization

I began this project with a data analysis, documented in the file Analysis.ipynb, which led to the following key insights:

The most important predictive characteristics:

1. **Previous campaign outcome (poutcome):**
   Previously successful customers have an extremely high conversion rate (65%)
   Even customers previously contacted unsuccessfully have an above-average conversion rate (14%)
2. **Contact method (contact):**
   Mobile phone contact (cellular) has a significantly higher conversion rate (14.7%) than landline phone (5.2%)
3. **Month of contact (month):**
   March, September, October and December have conversion rates 4-5 times above average (40-50%)
   Summer months have lower conversion rates
4. **Job:**
   Students (31.4%) and retirees (25.2%) have the highest conversion rates
   Blue-collar workers have the lowest conversion rates (6.9%)
5. **Education level (education):**
   People without formal education and those with higher education have above average conversion rates
6. **Factor combinations:**
   The combination "mobile phone + previous successful result + month Sep/Oct" has conversion rates of over 70%

Then, I started testing several classification machine learning models. I chose Precision, Recall, F1-score, and ROC-AUC as evaluation metrics. These metrics are important because the dataset is imbalanced, with only 11.27% positive cases, making Accuracy an irrelevant metric.

Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC
Gradient Boosting	**0.9209**	0.691	0.5379	0.6049	**0.9519**
XGBoost	0.8872	0.4995	0.8534	**0.6302**	0.944
Logistic Regression	0.8614	0.4438	**0.9095**	0.5966	0.9419
SVM	0.8442	0.4089	0.8586	0.5539	0.9249
Random Forest	0.9125	**0.702**	0.3879	0.4997	0.9458

According to the results in the table, **Gradient Boosting** appears to be the winning option. However, I will choose **XGBoost** :)

I chose this model because it has a high Recall, which is crucial in order to avoid missing positive cases, and its strong F1-Score, indicating a good balance between Precision and Recall.

XGBoost is a more suitable choice for production and future retraining because it offers:

- **Faster training** and better computational efficiency compared to standard Gradient Boosting.
- **Native support for incremental retraining**, allowing the model to be updated with new data without restarting from scratch.
- **Better handling of class imbalance**, which is essential when data distributions may shift over time.
- **Advanced regularization**, reducing the risk of overfitting on historical data.
- **Scalability and easy integration** into modern machine learning pipelines.

These features make XGBoost more robust and sustainable for long-term use.

Profit Optimization Pipeline for Call Campaigns

The implemented pipeline is an advanced predictive analytics system that optimally selects customers for Znailla Bank's call campaigns. Using XGBoost algorithms on historical data, the system identifies factors with maximum impact on conversion (contact channel, occupation, history) and generates a prioritized list of customers with their individual conversion probability. By applying a mathematically determined threshold, this pipeline maximizes net profit (considering the cost of €8 per call and profit of €80 per conversion), significantly reducing the number of calls and focusing efforts only on customers with real potential. The result is exported as a CSV file containing the list of recommended customers to call, along with their conversion probability score, thus optimizing resource allocation and maximizing campaign profitability.

euribor3m	nr.employed	y	conversion_	should_call
0.653	5008.7	yes	0.991463	1
0.642	5008.7	yes	0.99058646	1
0.846	4991.6	yes	0.9905173	1
0.881	4963.6	yes	0.9900827	1

Original Approach (Call Everyone):

- **Weekly calls**: ~4,000 customers
- **Cost**: €32,000 per week (4,000 calls × €8 per call)
- **Conversions**: ~451 customers (11.27% conversion rate)
- **Revenue**: €36,080 per week (451 conversions × €80 profit per conversion)
- **Net profit**: €4,080 per week (€36,080 - €32,000)
- **Profit margin**: 11.27%

Optimized Approach (Call Only High-Probability Customers):

- **Weekly calls**: ~840 customers (79% reduction)
- **Cost**: €6,720 per week (840 calls × €8 per call)
- **Conversions**: ~310 customers (36.9% conversion rate)
- **Revenue**: €24,800 per week (310 conversions × €80 profit per conversion)
- **Net profit**: €18,080 per week (€24,800 - €6,720)
- **Profit margin**: 72.9%

Annual Financial Impact:

- **Annual call reduction**: 164,320 fewer calls
- **Annual cost savings**: €1,314,560 (164,320 calls × €8 per call)
- **Annual conversion reduction**: 7,347 fewer conversions
- **Annual revenue reduction**: €587,760 (7,347 conversions × €80 profit)
- **Annual net profit improvement**: €726,800 (€1,314,560 savings - €587,760 revenue reduction)

Model Monitoring

1. **Performance Tracking**: We'll track key metrics weekly, focusing on:
   - Conversion rate of recommended customers
   - Model precision (% of called customers who convert)
   - Profit generated vs. costs
2. **Distribution Monitoring**: We'll compare the distribution of incoming customer data against our training data to detect significant changes that might affect model performance.
3. **Threshold Verification**: Periodically check if our profit-maximizing threshold remains optimal as business conditions change.

Model Retraining

1. **Scheduled Updates**: Retrain the model monthly with newly accumulated data to capture evolving customer behaviors and economic conditions.
2. **Trigger-Based Retraining**: Initiate retraining if:
   - Conversion rate drops below a defined threshold (e.g., 5% lower than expected)
   - Data distributions shift significantly
   - Business rules or product terms change
3. **Simple Validation**: Before deploying any retrained model, we'll:
   - Compare its performance against the current production model
   - Verify it still maximizes profit given the current cost structure
   - Keeping human review in the loop for major model changes