Final Project - Technical Report

Predicting Customer Churn

M.Sc. Business Analytics
Cloud Platforms (AWS)
Prof. Jordi Paliesse

April, 8th 2025

Team 3: Sophie Bald, Harshita Chivukula, Timo Sachs, Pau Soler, Darren Tan

Abstract

This report tackles a core business challenge in the telecom sector: predicting customer churn to support proactive retention strategies. Using a Telco Customer Churn dataset, we developed and deployed a scalable end-to-end machine learning pipeline in AWS, leveraging SageMaker for model training and inference. Our best-performing model achieved an ROC-AUC of 0.87 and was evaluated using a profit-based metric leading to over $300,000 in additional net revenue. Key churn drivers were identified, including contract type, service bundling, and customer tenure. Our solution demonstrates how cloud-based analytics can translate raw data into actionable insights with measurable financial impact.

Table of Contents

1. Business Proposal....1
2. Dataset description....2
3. Technical solution and key decisions....3
4. Evaluation of results....4
5. Reproduction instructions....5
Appendix....6

1. Business Proposal

Churn, in this case meaning a customer voluntarily cancelling their contract, leads to direct revenue loss for telecommunications companies and thus additional customer acquisition costs to gain new clients. Customer loyalty therefore is key for stable profitability and growth, so identifying how and why customers churn can help companies take proactive steps to retain them. This project analyzes customer churn for a telecommunications company using a comprehensive dataset containing demographic information, service subscriptions, location details, and churn behavior.

Therefore, the key business question we aim to address is: **How can we leverage our customer data to identify the key factors that predict churn, enabling us to implement timely and effective retention measures?**

The insights derived from this analysis will help the business reduce customer attrition, develop actionable retention strategies, increase customer lifetime value, and drive profitability. To estimate the potential economic benefit of our machine learning model, we decided to evaluate a marketing strategy targeting customers predicted to churn. First, we define outcomes and associated financial impacts. We assume the following numbers are partially based on previous experiences of the companies marketing departments.

- **True positive profit:** An identified churner targeted with the retention campaign (60% of average Customer Lifetime Value minus Customer Acquisition Cost).
- **False positive cost: a non-churner unnecessarily targeted**
  Customer Acquisition Cost (defined as 10% of average CLTV) because we spend money as part of the campaign to retain a customer that would have stayed either way.
- **True negative** (disregarded)**: a non-churner correctly identified**
  0 because the customer stays and we do not incur campaign costs.
- **False negative** (disregarded)**: a churner incorrectly identified as a loyal customer**
  0 because this is a lost opportunity and we do not assume direct campaign costs.

As such, our profit function can be described as:

$$Profit = TP * (\varnothing\ CLTV * 0.6 - \varnothing\ CLTV * 0.1) - FP * (\varnothing\ CLTV * 0.1)$$

This function is used later during (offline) model training as a custom scoring metric using the `make_scorer` function from sklearn. Unfortunately it is not available in Sagemaker, so we used the area-under-the-curve maximization function. In the end, both approaches led to the same maximum profit for the telecommunications company as we shall see later.

2. Dataset description

This project utilizes the Telco Customer Churn dataset from Hugging Face, which provides a comprehensive view of customer attributes, service usage, location data, and churn behavior. Through targeted feature engineering and reduction, the dataset was cleaned and prepared for modeling to predict customer churn more effectively.

- **Customer Identification & Demographics:** Core attributes like Age, Gender, and Number of Dependents were retained to describe the customer base. Dropped columns include geographical variables and other non-predictive demographic variables.
- **Service & Contract Details:** Key features such as Contract type, Internet Service, Internet Type, and various binary service indicators were kept and appropriately encoded. Contract and Internet Type were treated as ordinal features. Columns that were used in feature engineering and those that were highly correlated to other columns were removed to prevent redundancy.
- **Financial Metrics:** Most financial columns were dropped. Total Extra Data Charges was retained as a useful predictor.
- **Usage & Behavior:** Usage data such as Avg Monthly GB Download, Avg Monthly Long Distance Charges, and Tenure in Months were preserved to reflect customer engagement over time.
- **Churn Information:** Churn was retained as the binary target variable for modeling. Other churn-related columns were excluded to avoid post-outcome bias.
- **Time Information:** The Quarter column was dropped as it didn’t meaningfully contribute to churn prediction.

Additionally, after our first iteration of the model and going through the entire machine learning lifecycle, we made the decision to drop the Satisfaction Score as it was causing data leakage and unrealistically good model results (~99% accuracy & ROC-AUC score).

We also decided to add newly engineered features which include:

- **Refunds Awarded:** A new binary feature was created to indicate whether a customer received any refunds. Customers with Total Refunds greater than zero were assigned a 1, and the rest were assigned a 0.
- **Monthly charge/CLTV:** This new ratio feature gives us a normalized measure that can be compared across different customer segments, regardless of the absolute values of monthly charges or CLTV. This makes it easier to identify patterns, e.g. for customers who spend relatively more than their expected CLTV and are thus likely to churn.
- **Phone Bundle:** A binary feature was created to flag customers who subscribe to all four key phone-related services: Phone Service, Multiple Lines, Premium Tech Support, and Device Protection Plan. Customers using all four were classified as having a full phone bundle.

- **Service Level:** Customers were grouped into Low, Medium, or High service levels based on the total number of services used. These categories were created by binning the total count of subscribed services to capture levels of engagement.

After completing the data preprocessing steps, all numeric features were standardized for consistency. In summary, out of 49 original features, we retained 25 predictive features, dropped 24 features, and engineered 4 new features to ensure our dataset is clean, focused, and optimized for building a robust churn prediction model.

3. Technical solution and key decisions

Our technical solution leverages the AWS cloud platform to build a robust, end-to-end pipeline for predicting customer churn, focusing on efficiency, scalability, and cost-effectiveness. We describe the overall workflow in the figure below.

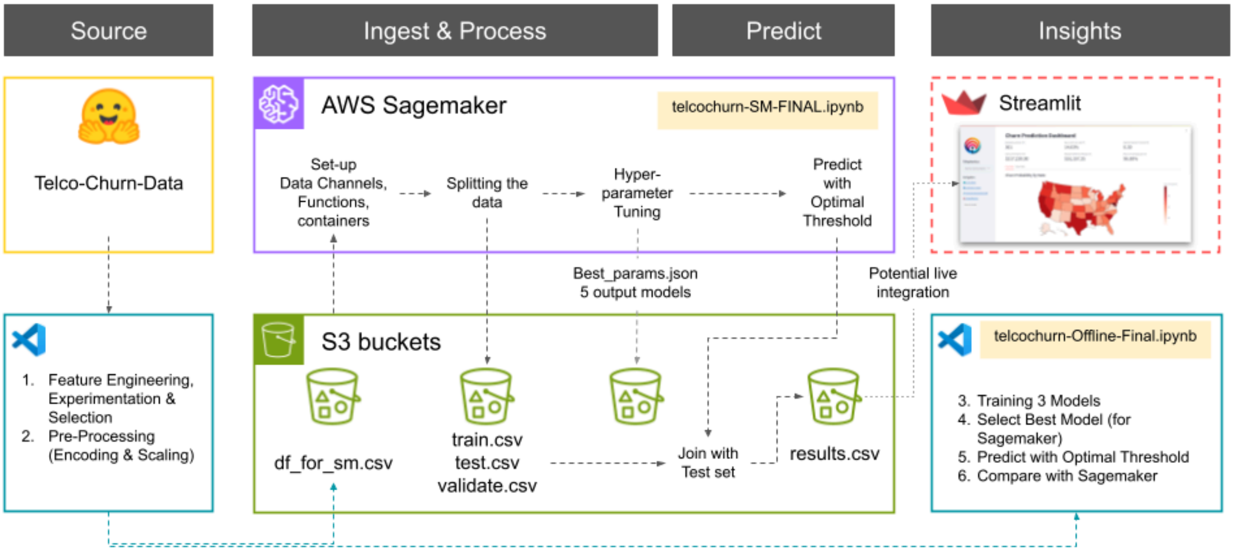

Source
Ingest & Process
Predict
Insights
Telco-Churn-Data
AWS Sagemaker
telcochurn-SM-FINAL.ipynb
Set-up Data Channels, Functions, containers
Splitting the data
Hyper-parameter Tuning
Predict with Optimal Threshold
Streamlit
Best_params.json
5 output models
Potential live integration
1. Feature Engineering, Experimentation & Selection
2. Pre-Processing (Encoding & Scaling)
S3 buckets
df_for_sm.csv
train.csv
test.csv
validate.csv
Join with Test set
results.csv
telcochurn-Offline-Final.ipynb
3. Training 3 Models
4. Select Best Model (for Sagemaker)
5. Predict with Optimal Threshold
6. Compare with Sagemaker

Figure 1: AWS Workflow - Key components, Decisions and Choices in Solution

Feature Encoding and Scaling: Ordinal features such as Contract, Internet Type, and Service Level are encoded and scaled to preserve their natural order and ensure comparability. These preprocessing steps are done in the offline notebook to output a dataframe for final use in Sagemaker.

Handling Data Leakage: To avoid leakage, we dropped the feature 'Satisfaction Score' that could reflect outcomes which would only be known *after* the contract has already been cancelled; not doing so would result in an unrealistically high AUROC score of 0.992.

Model Training and Evaluation:
To avoid excessive charges from unnecessary model training in Sagemaker, the training and evaluation of 3 models (Linear Regression, Random Forest, and XGBoost) were performed on

the offline notebook to select the best model. The best model trained offline would then be trained in Sagemaker, utilizing the unique hyperparameter-tuning workflows of Sagemaker. Cross-validation (using an 80-10-10 split) in both the offline notebook and Sagemaker ensured robust model evaluation and comparison of the tuning flow and eventual predictions.

Deployment and Inference:
The selected XGBoost binary classification model is deployed as a SageMaker endpoint for real-time inference, using the test set to generate predictions (10% of the dataset of approximately 7040 rows). First, the imported best threshold (e.g., 0.30) from our offline notebook was used. Then, the batch size of 704, corresponding to the entire test set, is configured to optimize processing. Real-time inference is configured to run cost-effectively so that endpoints are open and closed to manually make predictions in real time. This avoids extra charges from using batch transform jobs to make predictions. Workaround functions are defined to allow the inference of predictions.

4. Evaluation of results

As outlined in the introduction, **our goal is to leverage the customer data to identify the key factors that predict churn.** The outlined pipeline we created thus actually would enable the telco company to implement timely and effective retention measures. Using the profit scoring metric defined in the first chapter ensures that the model's performance translates to real dollar value impact. While Sagemaker does not support custom evaluation functions for profit maximization, we developed our own functions while also using Sagemaker's built-in 'validation:auc' metric. Ultimately, both approaches led to the same maximum profit of $308,021.

Overall, the trained XGBoost model demonstrated a really good performance in both offline and online training, achieving the highest profit and an impressive recall of over 95%. Regarding the feature importance, contract type emerged as the most significant factor, accounting for 36.65% of churn prediction, followed by Internet Service Type (13.96%), Number of Dependents (10.76%), Monthly charge/CLTV Ratio (8.99%), and Customer Tenure (5.25%). These feature importances are consistent across offline and online training as seen in Appendix 1 and 2.
The analysis suggests a focus on converting customers to longer-term contracts while optimizing internet service offerings. Therefore, developing family-focused retention programs and implementing dynamic pricing based on customer value will be crucial. Additionally, creating targeted loyalty programs for long-term customers should help strengthen customer relationships.

To maximize the impact of this pipeline and model, we recommend deploying an **automated system for early risk detection**. Resources should be concentrated on high-value, high-risk customers, with a continuous monitoring and retention strategy in place. One such system can be found in Appendix 3 or at churn.timosachs.de, where we created a Streamlit dashboard connected to a churn dataset to visualize churn risk and enable customer service agents to take immediate action by sending LLM-based retention emails.

5. Reproduction instructions

In this final section we will focus on the process followed for the online Sagemaker Notebook, used for final results and evaluation to ensure replicability. The process for the offline data preprocessing notebook can be found in appendix 4 as well as the file structure in Sagemaker in Appendix 5. For easier reading, we decided to keep the steps in a numbered bullet point like format.

Section 1: Setting up the Sagemaker Environment

1. Load the processed dataframe **'df_for_sm.csv'** and the 2 pickle files **'average_cltv.pkl**' and **'best_threshold.pkl'**.
2. Define bucket and folder architecture for input and output of files.
3. To reduce costs, we avoided using the batch transform job service provided by AWS, but instead defined a workaround function for real-time inference of predictions for our test set data, creating significant cost savings (~€0.40 instead of €2 per batch job).
4. Create train-test-validation split data files to be retrieved as objects by S3 in their respective folders (**'train', 'test', validate'**), following an 80-10-10 train-test-validation split ratio by the default Sagemaker behaviour
5. Define binary conversion function from the best threshold imported from the offline notebook, obtained using custom metric 'profit' that is not available in Sagemaker
6. Define the algorithm's containers and the input data channels for training and validation

Section 2: Hyperparameter Tuning for Selected Best Model XGBoost

1. Define hyper-paramers and ranges suitable for XGBoost model, using AUC as the defined metric as a workaround to our custom metric 'profit'
2. Create and run 5 tuning jobs
3. Store the 5 hyperparameter-trained models into **'output'** folder
4. Analyze tuning results and select the best model determined by Sagemaker

Section 3: Endpoint Deployment

1. Create an endpoint for prediction and select only the best model trained in Section 2
2. Make predictions into 'batch-in.csv' stored in **'batch-out'** folder
3. Convert to binary predictions for model evaluation and metrics based on the best threshold from the offline notebook
4. Re-optimize thresholds in Sagemaker using the main metric as AUC

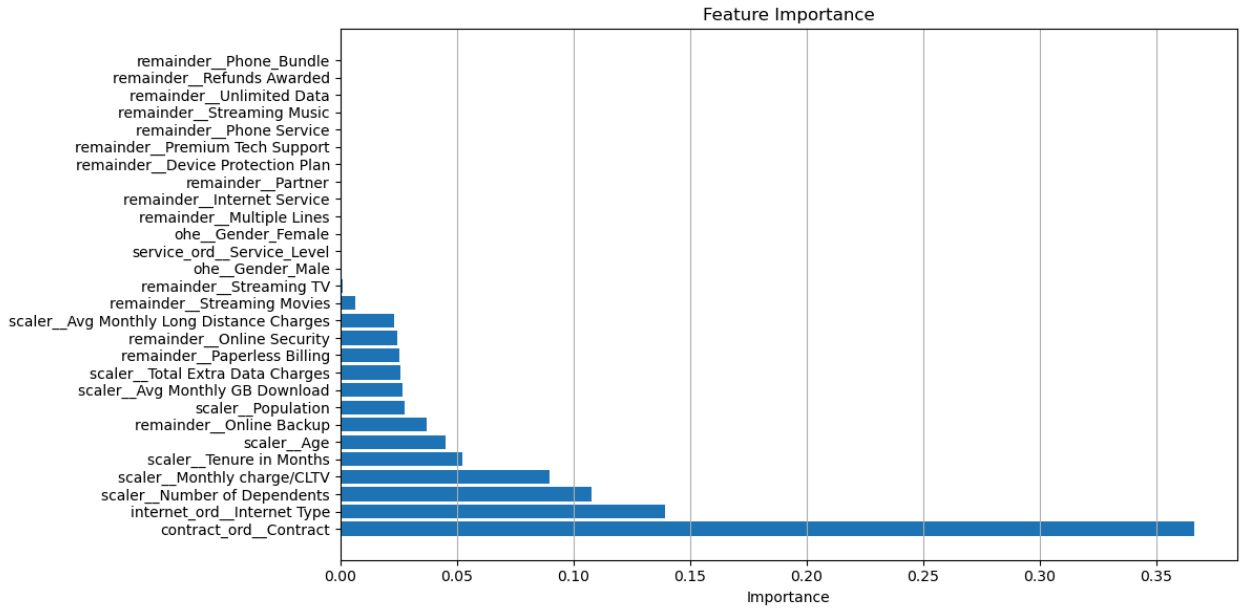
Section 4: Cleanup and Save Predictions

1. Compare the offline notebook and Sagemaker's best thresholds respectively (as **'best_threshold.pkl'** and **'best_threshold_new.pkl'**), and confirm there is no/minimal difference in optimal profits at each best threshold
2. Save the best model to the 'model' folder, and the resulting best predictions onto the test set dataframe as 'prediction_results.csv' to the 'results' folder for evaluation
3. Delete endpoint and check to prevent additional unwanted charges

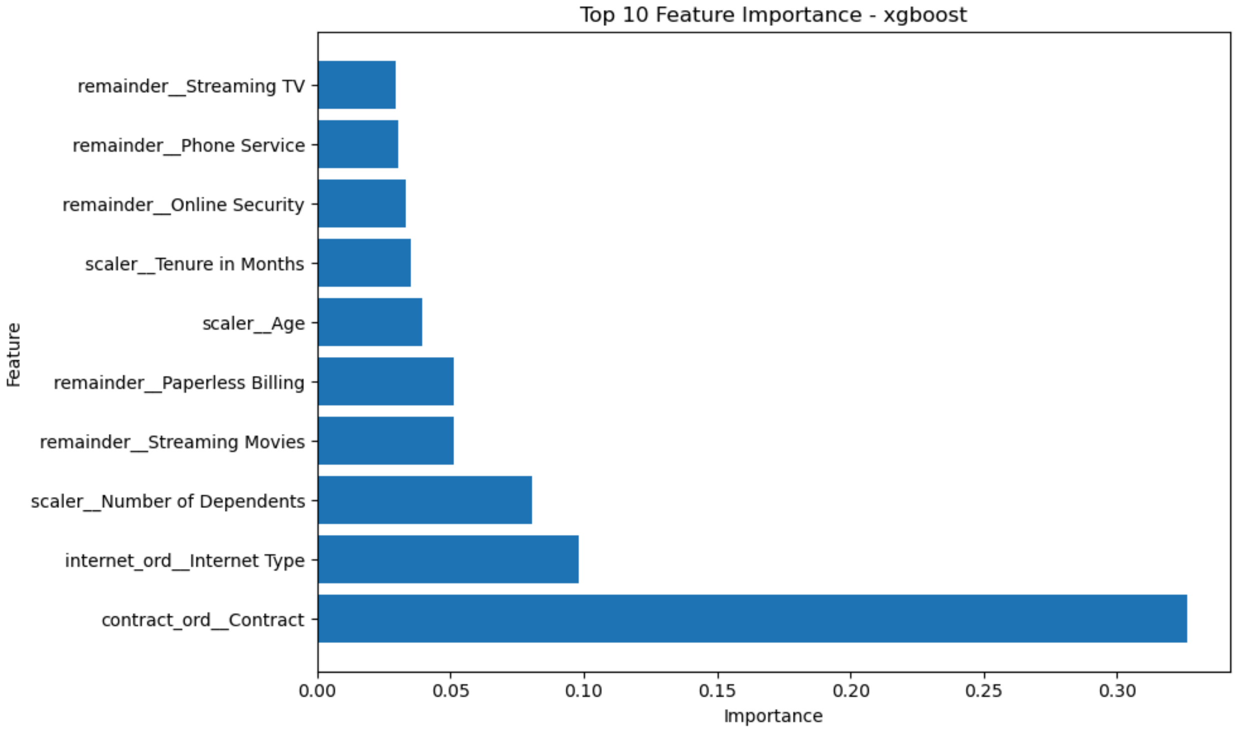
Appendix

Appendix 1: Best Model Feature Importance (Sagemaker)..7
Appendix 2: Best Model Feature Importance (Offline Training)...7
Appendix 3: Prototype for churn risk detection and intervention... 8
Appendix 4: Process offline notebook used for pre-processing and feature engineering..............9
Appendix 5: Structure of Output Folders in S3 buckets and GitHub repo.................................. 10

Appendix 1: Best Model Feature Importance (Sagemaker)

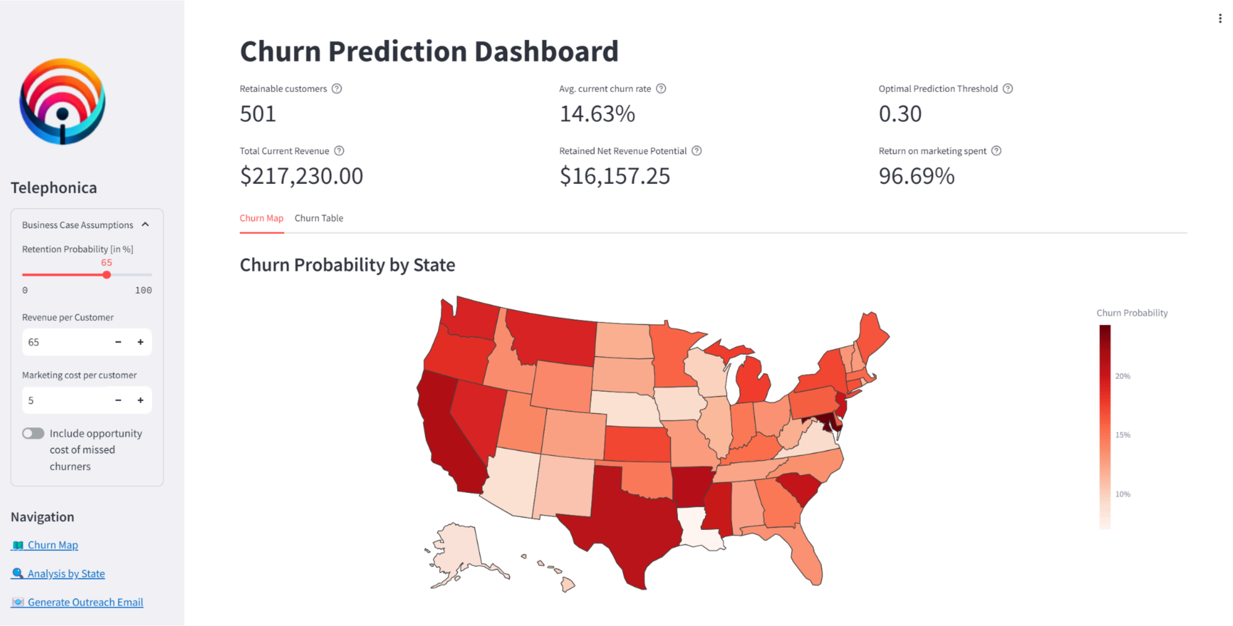


Appendix 2: Best Model Feature Importance (Offline Training)



Appendix 3: Prototype for churn risk detection and intervention

Here the location information is added back to the dataset to visualize churn risk per US state to further deep-dive into location-specific reasons. The production version of this dashboard, although with a slightly different underlying dataset can be found here: churn.timosachs.de.



Telephonica

Business Case Assumptions

Retention Probability [in %]: 65 (0–100)

Revenue per Customer: 65

Marketing cost per customer: 5

Include opportunity cost of missed churners

Navigation

- Churn Map
- Analysis by State
- Generate Outreach Email

Churn Prediction Dashboard

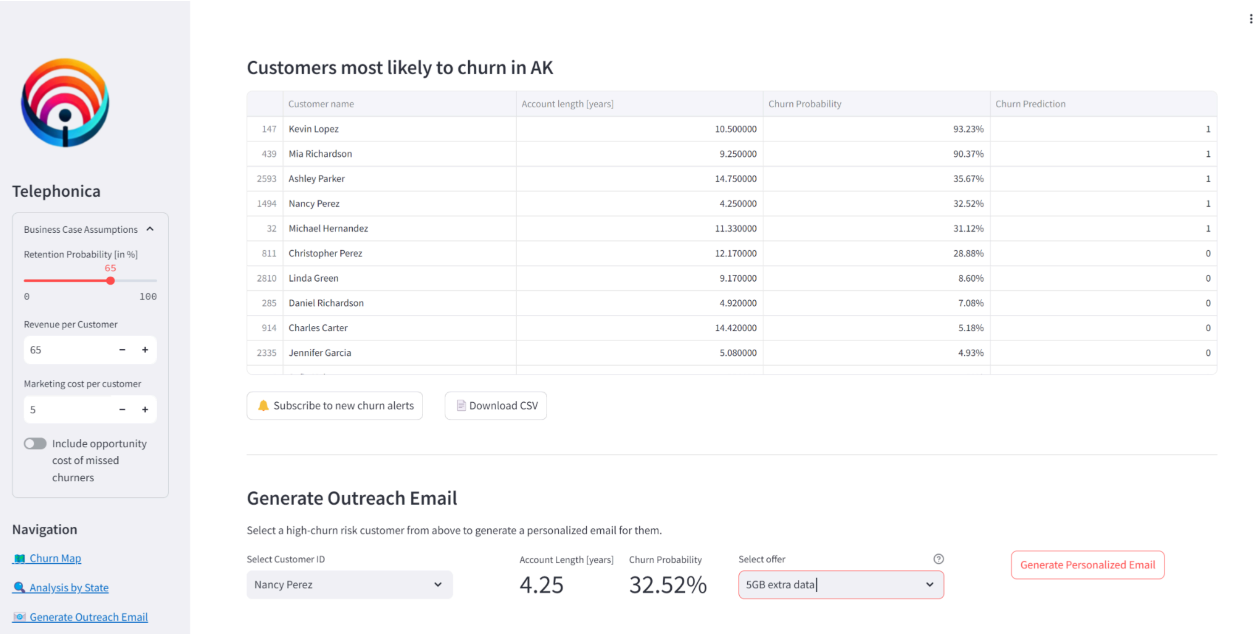
Retainable customers	Avg. current churn rate	Optimal Prediction Threshold
501	14.63%	0.30

Total Current Revenue	Retained Net Revenue Potential	Return on marketing spent
$217,230.00	$16,157.25	96.69%

Churn Map | Churn Table

Churn Probability by State

In this prototype, customer service agents can interact with a dashboard to view churn risk per state and take actions accordingly. Customer service agents could then have the option, in a next step, to get a list of the top e.g. 10 customers most likely to churn per state and send a preemptive email offering loyalty incentives to prevent the customer from churning.



Telephonica

Business Case Assumptions

Retention Probability [in %]: 65 (0–100)

Revenue per Customer: 65

Marketing cost per customer: 5

Include opportunity cost of missed churners

Navigation

- Churn Map
- Analysis by State
- Generate Outreach Email

Customers most likely to churn in AK

	Customer name	Account length [years]	Churn Probability	Churn Prediction
147	Kevin Lopez	10.500000	93.23%	1
439	Mia Richardson	9.250000	90.37%	1
2593	Ashley Parker	14.750000	35.67%	1
1494	Nancy Perez	4.250000	32.52%	1
32	Michael Hernandez	11.330000	31.12%	1
811	Christopher Perez	12.170000	28.88%	0
2810	Linda Green	9.170000	8.60%	0
285	Daniel Richardson	4.920000	7.08%	0
914	Charles Carter	14.420000	5.18%	0
2335	Jennifer Garcia	5.080000	4.93%	0

Generate Outreach Email

Select a high-churn risk customer from above to generate a personalized email for them.

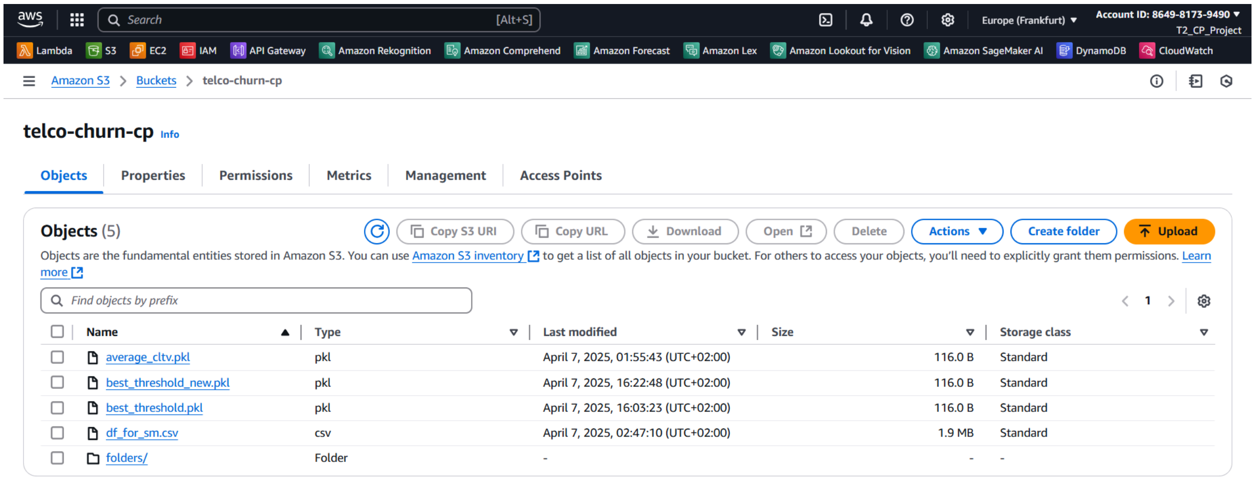
Select Customer ID	Account Length [years]	Churn Probability	Select offer
Nancy Perez	4.25	32.52%	5GB extra data

Appendix 4: Process offline notebook used for pre-processing and feature engineering

An experimental process continuing from Section 2 in the Jupyter Notebook:

1. Calculated average CLTV that would be required for our scoring matrix and saved into a pickle file as **‘average_cltv.pkl**’.
2. Preprocessed the data
3. Moved the target variable column ‘Churn’ to the first index column for Sagemaker’s predictions to work
4. Stored the preprocessed dataframe into a csv file as **‘df_for_sm.csv’.**
5. Set split = 10% of the entire dataset to be aligned with Sagemaker’s 80-10-10 split for verification purposes
6. Used 3 models with hyperparameters: Linear Regression, Random Forest and XGBoost, accounting for mild class imbalance with scale_pos_weight.
7. Set metric to optimize as Profit (unable to do so in Sagemaker - workbook used AUC first)
8. Determined that **XGboost with hyperparameter tuning is the best model to be trained in Sagemaker**
9. Plotted feature importance graph to check for potential data leakage.
    - Discovered that with the variable ‘Customer Satisfaction’, it contributes to the AUROC score being = 0.992.
    - It is conjectured that Customer Satisfaction would be a result that is known after a Telco service is delivered, which happens concurrently with Churn behaviour.
10. Optimized decision threshold based on profit as main metric, and saved the best threshold into **‘best_threshold.pkl’**.
11. Repeat Steps 2 to 10, experimenting with the notion that the ‘Customer Satisfaction’ variable contributes to leakage.

Appendix 5: Structure of Output Folders in S3 buckets and GitHub repo

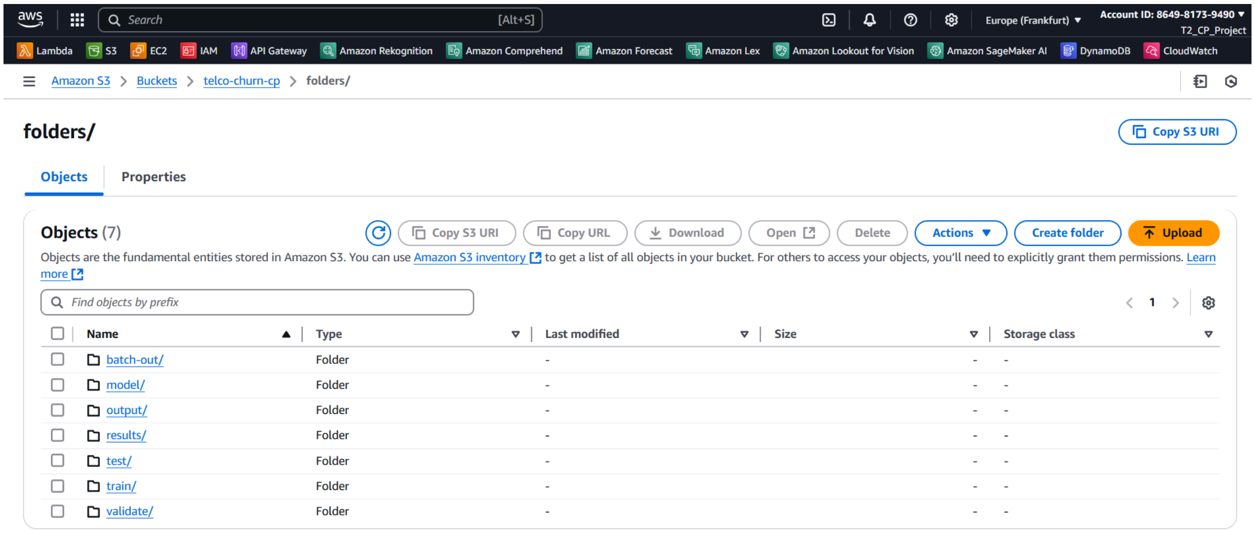

aws
Search [Alt+S]
Europe (Frankfurt)
Account ID: 8649-8173-9490
T2_CP_Project
Lambda S3 EC2 IAM API Gateway Amazon Rekognition Amazon Comprehend Amazon Forecast Amazon Lex Amazon Lookout for Vision Amazon SageMaker AI DynamoDB CloudWatch
Amazon S3 > Buckets > telco-churn-cp
Info
Objects Properties Permissions Metrics Management Access Points
Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload
Find objects by prefix
1

telco-churn-cp

Objects (5)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
average_cltv.pkl	pkl	April 7, 2025, 01:55:43 (UTC+02:00)	116.0 B	Standard
best_threshold_new.pkl	pkl	April 7, 2025, 16:22:48 (UTC+02:00)	116.0 B	Standard
best_threshold.pkl	pkl	April 7, 2025, 16:03:23 (UTC+02:00)	116.0 B	Standard
df_for_sm.csv	csv	April 7, 2025, 02:47:10 (UTC+02:00)	1.9 MB	Standard
folders/	Folder	-	-	-

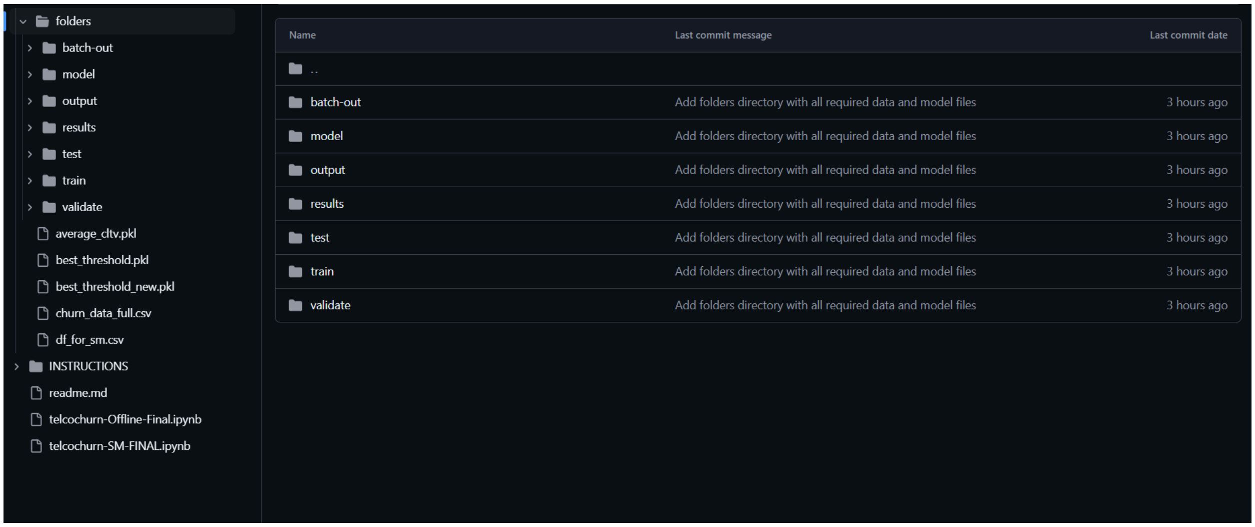

aws
Search [Alt+S]
Europe (Frankfurt)
Account ID: 8649-8173-9490
T2_CP_Project
Lambda S3 EC2 IAM API Gateway Amazon Rekognition Amazon Comprehend Amazon Forecast Amazon Lex Amazon Lookout for Vision Amazon SageMaker AI DynamoDB CloudWatch
Amazon S3 > Buckets > telco-churn-cp > folders/
Copy S3 URI
Objects Properties
Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload
Find objects by prefix
1

folders/

Objects (7)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
batch-out/	Folder	-	-	-
model/	Folder	-	-	-
output/	Folder	-	-	-
results/	Folder	-	-	-
test/	Folder	-	-	-
train/	Folder	-	-	-
validate/	Folder	-	-	-



- folders
  - batch-out
  - model
  - output
  - results
  - test
  - train
  - validate
  - average_cltv.pkl
  - best_threshold.pkl
  - best_threshold_new.pkl
  - churn_data_full.csv
  - df_for_sm.csv
- INSTRUCTIONS
- readme.md
- telcochurn-Offline-Final.ipynb
- telcochurn-SM-FINAL.ipynb

Name	Last commit message	Last commit date
..		
batch-out	Add folders directory with all required data and model files	3 hours ago
model	Add folders directory with all required data and model files	3 hours ago
output	Add folders directory with all required data and model files	3 hours ago
results	Add folders directory with all required data and model files	3 hours ago
test	Add folders directory with all required data and model files	3 hours ago
train	Add folders directory with all required data and model files	3 hours ago
validate	Add folders directory with all required data and model files	3 hours ago