TCCP Technical Paper

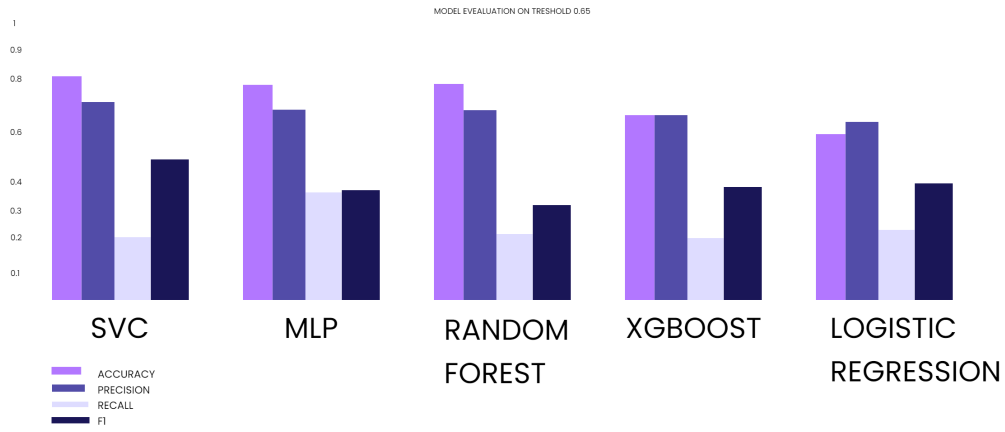
Mikiyas Zenebe Selemun Abrha Yafet Mulaw Haile Sintayehu Fisehatsion Adisu Tedros Nigus

TCCP

Trustworthy Customer Churn Prediction

From customer data to governed retention action

A reproducible machine-learning, explainability, and MLOps system



IBM Telco Customer Churn Dataset

MLflow • Model Registry • FastAPI • CI/CD

August 2026

Abstract

Churn prediction becomes valuable only when it supports a safe and explainable decision. TCCP is an end-to-end system that turns the IBM Telco Customer Churn dataset into calibrated customer-risk estimates, interpretable drivers, and a ranked retention action list. The system cleans the source data, compares five classifiers, selects an RBF support-vector classifier (SVC) using validation PR-AUC, evaluates discrimination and calibration on a held-out test set, and converts risk into expected net value under a contact budget. The final model achieved test ROC-AUC = 0.846, PR-AUC = 0.667, and Brier score = 0.137. The implementation also packages feature contracts, batch scoring, model registry stages, an API, and CI/CD controls so that a prediction can be traced from input data to production action.

Project scope and challenge level. The implemented version uses the IBM Telco Customer Churn dataset from Kaggle, with approximately 7,000 customer records. The build objective is a churn model whose predicted probabilities are trustworthy enough to drive a retention budget, together with a SHAP-based report of churn drivers and a top-k targeting simulation. The intermediate difficulty comes from making a budget decision under expected value, rather than reporting AUC alone. Categorical encoding, calibration curves, Brier score, and SHAP are therefore core methods. The WSDM KKBox Churn dataset, which contains millions of records across separate transaction and user-log tables, is reserved as the harder extension and is not the dataset used for the results reported in this paper.

1. TCCP problem statement

Telecom churn is not only a classification problem. A business team must decide which customers to contact, what evidence supports that decision, and how many contacts the budget can afford. A model that maximizes accuracy while producing poorly calibrated probabilities can waste offers on low-risk customers or miss customers who need intervention.

TCCP defines the operational problem as follows. For a customer with feature vector x, estimate the probability of churn:

$$p(x) = P(Y = 1 \mid X = x), \qquad Y \in \{0, 1\}, \tag{1}$$

where $Y = 1$ means that the customer churns. The system then ranks customers using expected intervention value:

$$U(x) = p(x)\, r\, v - c, \tag{2}$$

where r is the assumed save rate, v is customer value, and c is contact cost. Under a finite budget, customers are selected from the highest-ranked risk until the contact limit is reached. This separation between prediction and action is central: model quality determines ranking, while business assumptions determine intervention.

Research question. Can a reproducible pipeline produce reliable churn probabilities, explain why risk changes, and allocate retention contacts transparently under a fixed budget?

2. Solution architecture

TCCP follows a problem-to-solution chain:

Business problem		**TCCP solution**
Messy customer records	→	Leakage-safe cleaning, numeric coercion, imputation, encoding, and a versioned 32-feature contract.
Risk scores are hard to compare	→	Five-model benchmark, validation PR-AUC selection, held-out evaluation, and calibration diagnostics.
Black-box scores are difficult to trust	→	SHAP global explanations, SHAP dependence plots, partial dependence, and permutation importance.
Retention budget is limited	→	Probability ranking, action tiers, and expected net value by contact budget.
Models drift or become untraceable	→	Serialized artifact, metadata, registry stages, batch scoring, API health checks, and CI/CD validation.

The design principle is traceability. Every production score should be reproducible from the model version, feature schema, input record, and scoring code. The system therefore treats data quality, probability quality, explainability, and deployment governance as one product rather than separate notebooks.

3. Data understanding and preparation

The raw IBM Telco Customer Churn file contains 7,043 customers. The cleaning stage strips column names, converts the whitespace-encoded `TotalCharges` field to numeric values, maps `Churn` from Yes/No to 1/0, removes `customerID` from model inputs, and separates target from predictors. Numeric fields are median-imputed and standardized; categorical fields are most-frequent-imputed and one-hot encoded inside a scikit-learn pipeline. Engineered variables include service count, tenure bins, charge interactions, charge-per-tenure, logarithmic charges, and a new-customer flag. The dataset source is the publicly available IBM Telco Customer Churn dataset on Kaggle.

The code also includes an optional KKBox preparation path in `src/kkbox_features.py`. KKBox is not treated as if it had the same row structure as Telco. Its `train` labels are joined to `members`, while `transactions` and `user_logs` are aggregated by `msno` in chunks. This produces one modelling row per subscriber and avoids counting repeated event rows as independent customers. The resulting flat file uses the same `Churn` target and can be passed to the common training script. The current reported metrics remain Telco results; KKBox requires a separate run because its label definition, class balance, temporal split, and feature distributions differ.

Data contract item	Observed implementation
Source	IBM Telco Customer Churn dataset; Kaggle source link
Rows retained	7,043 customers
Raw missing values	11 whitespace values in `TotalCharges`
Cleaned feature count	32 columns
Target	`Churn`: 1 = churn, 0 = stay
Identifiers	`customerID` excluded before training
Split	80/20 stratified holdout
Random seed	42

Table 1: TCCP data and reproducibility contract.

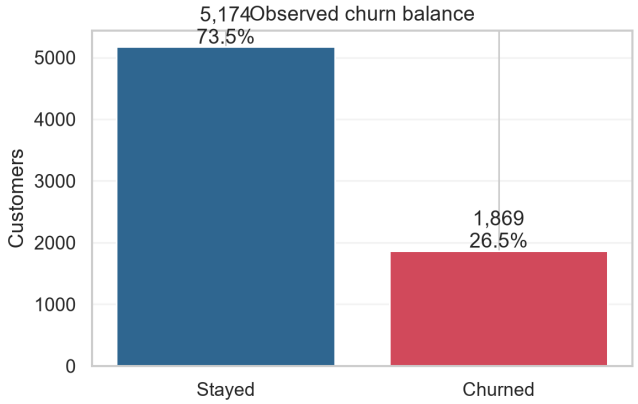


Figure 1: Observed churn balance in the cleaned dataset.

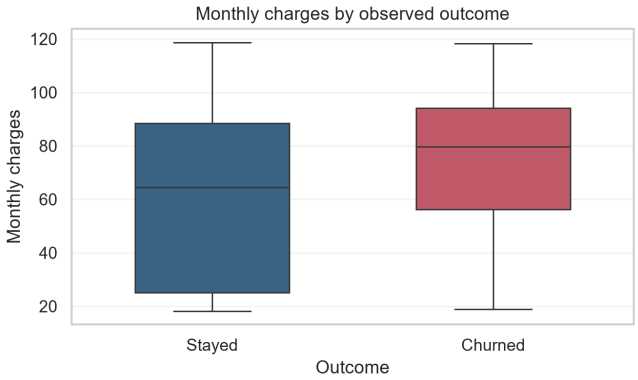


Figure 2: Monthly charges by observed churn outcome.

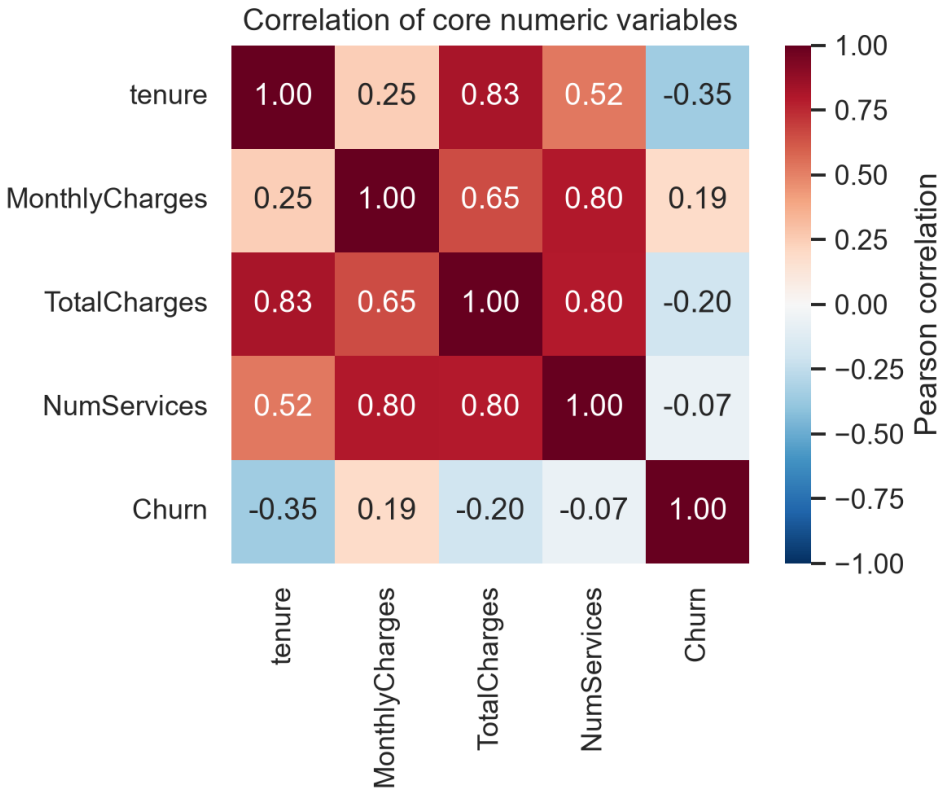


Figure 3: Correlation of core numeric variables in the cleaned 32-feature dataset.

4. Modelling and evaluation protocol

TCCP benchmarks five models: elastic-net logistic regression, RBF SVC, random forest, XGBoost, and a PyTorch multilayer perceptron. Hyperparameters are tuned only on the training partition using five-fold validation PR-AUC. The independent test set is used once for final reporting. This prevents test-set performance from silently becoming a model-selection criterion.

The primary selection metric is PR-AUC because churn is an imbalanced positive class and the operational team needs a useful ranking of likely churners. For non-technical readers, AUC is a summary of how well the model separates customers who churn from customers who stay: a value closer to 1.0 is better than a value near 0.5. ROC-AUC measures separation across all decision thresholds, while PR-AUC focuses on how useful the ranked high-risk group is. Accuracy describes the share of correct yes/no decisions at one threshold, and Brier score measures how close the predicted percentages are to the observed outcomes; lower Brier scores are better. The selected classifier is wrapped in five-fold sigmoid calibration, with the calibration folds kept inside the training partition. The independent test set is then used for the final reliability curve and Brier score, avoiding calibration leakage.

KKBox harder-version workflow

The repository also provides a reproducible KKBox extension using the same modelling family and decision objective. The one-row-per-customer Telco structure cannot be applied directly to KKBox because KKBox separates the label table, member table, transaction history, and daily listening logs. The KKBox preparation script first downloads or reads the four CSV tables, identifies the subscriber label in `train.csv` or `train_v2.csv`, joins member attributes, and aggregates repeated transaction and user-log records by `msno`. Transaction features include payment-plan, price, renewal, cancellation, count, and date-recency summaries. Listening features include active-log days, listening counts, unique songs, total seconds, and completion-level counts. The output is one row per subscriber with the common `Churn` target.

The source used for the extension is the publicly available WSDM KKBox Churn dataset, distributed through Kaggle. The dataset should be downloaded according to the provider's terms and kept outside version control because of its size and licensing conditions.

The KKBox experiment then uses the same five algorithms: elastic-net logistic regression, RBF SVC, random forest, XGBoost, and PyTorch MLP. The same evaluation outputs are retained: validation PR-AUC, ROC-AUC, PR-AUC, calibration curves, Brier score, precision, recall, F1, and top-k retention value under a fixed contact budget. The KKBox benchmark is run separately from the Telco benchmark because its churn definition, class balance, event

history, and temporal structure are different. In particular, a production KKBox study should use a time-based holdout aligned with the prediction period; random splitting is retained only as a quick Colab demonstration.

The complete Colab process is documented in `notebooks/02_model_training.ipynb`: install dependencies, set `TCCP_DATA_PATH` to the generated `kkbox_features.csv`, run the fast Colab benchmark, tune the five models, calibrate probabilities, and save the benchmark summary and model bundle. The preparation command is:

```
python -m src.kkbox_features --data-dir data/kkbox \\
  --output data/kkbox_features.csv --chunksize 250000
```

The current numerical tables and conclusions in this paper remain based on the IBM Telco experiment. KKBox results should be added only after the full benchmark has completed on the downloaded KKBox files.

Model	Validation PR-AUC	Test ROC-AUC	Test PR-AUC	Accuracy	Balanced accuracy
SVC (RBF)	0.670 ± 0.019	0.846 ± 0.011	0.667 ± 0.025	0.755	0.759
Logistic regression	0.670 ± 0.016	0.848 ± 0.011	0.665 ± 0.025	0.760	0.763
Neural network (MLP)	0.668 ± 0.016	0.846 ± 0.011	0.653 ± 0.026	0.779	0.757
XGBoost	0.666 ± 0.021	0.844 ± 0.011	0.657 ± 0.025	0.774	0.753
Random forest	0.665 ± 0.020	0.845 ± 0.011	0.655 ± 0.026	0.768	0.769

Table 2: Benchmark results on the independent test set. Uncertainty values are bootstrap standard deviations where available. The SVC is the TCCP champion by validation PR-AUC.

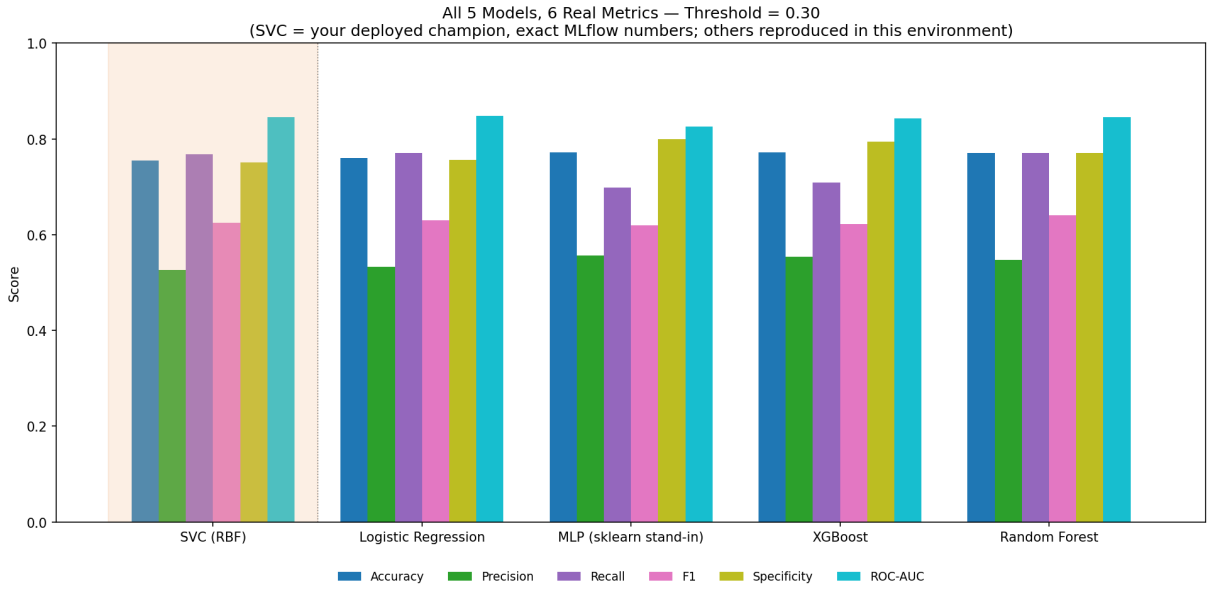


Figure 4: All five models across six reported metrics at threshold 0.30, as shown in the local project figure.

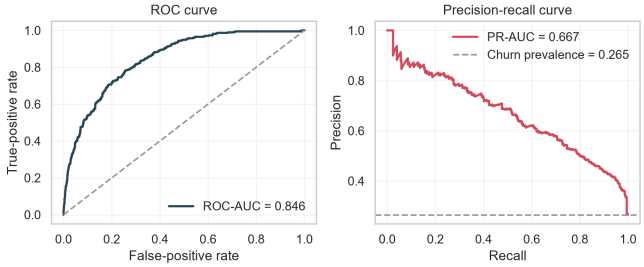


Figure 5: ROC and precision-recall curves from the held-out test predictions.

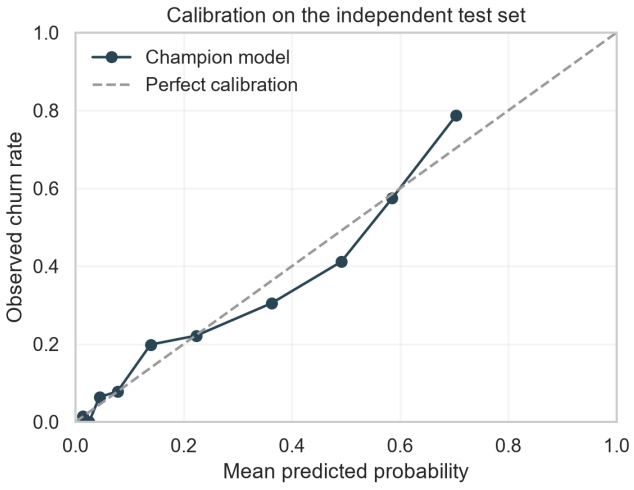


Figure 6: Reliability curve from ten quantile bins on the independent test set.

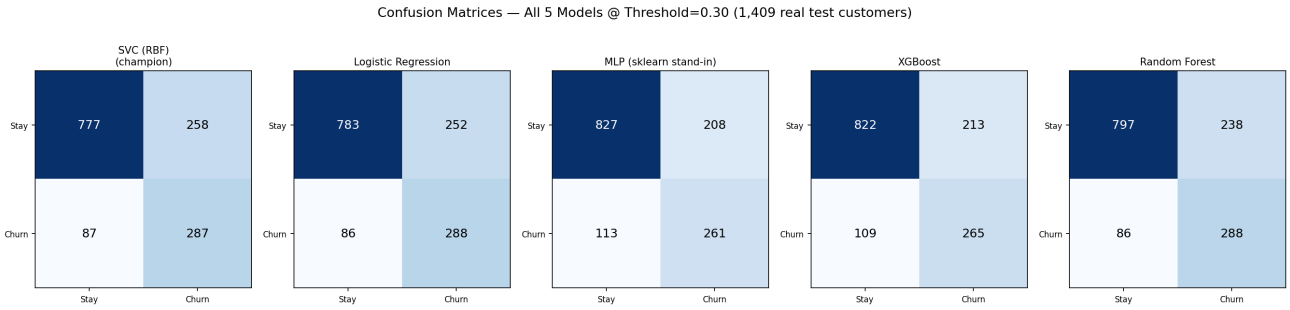


Figure 7: Confusion matrices for all five models at threshold 0.30 on 1,409 real test customers.

The calibration plot is close to the diagonal through most of the observed risk range, indicating that the predicted probabilities are usable as risk estimates rather than only as a ranking score. The highest-risk bin is above the diagonal, so the champion is somewhat conservative at the upper end: observed churn is higher than the mean predicted probability in that bin. This supports ranking for retention, but it does not remove the need to monitor calibration after deployment. The test Brier score of 0.137 summarizes the combined probability error across all customers.

5. Explainability: from score to reason

TCCP uses explanations as decision support, not as causal claims. SHAP values describe how the fitted model moves an individual prediction relative to a background sample. Positive SHAP values increase predicted churn output; negative values reduce it. Partial dependence shows the model's average response when one feature is varied while other rows are retained. These views answer different questions: SHAP explains local and global model behavior, while partial dependence summarizes an average response curve.

The available explainability images in `figures/files` are explicitly labeled as XGBoost diagnostics, while the selected production champion is the RBF SVC. They are therefore presented as real project diagnostics for the engineered feature set, not as SVC-specific measurements. This distinction keeps the visual evidence aligned with the model name printed inside each image.

The explanations support actionable hypotheses. In the local SHAP figures, Contract, tenure, Fiber_x_Charge, electronic check payments, and InternetService_Fiber are among the prominent displayed drivers. These are model associations, not proof that changing a contract, payment method, or service will cause a customer to stay; that question requires randomized intervention measurement.

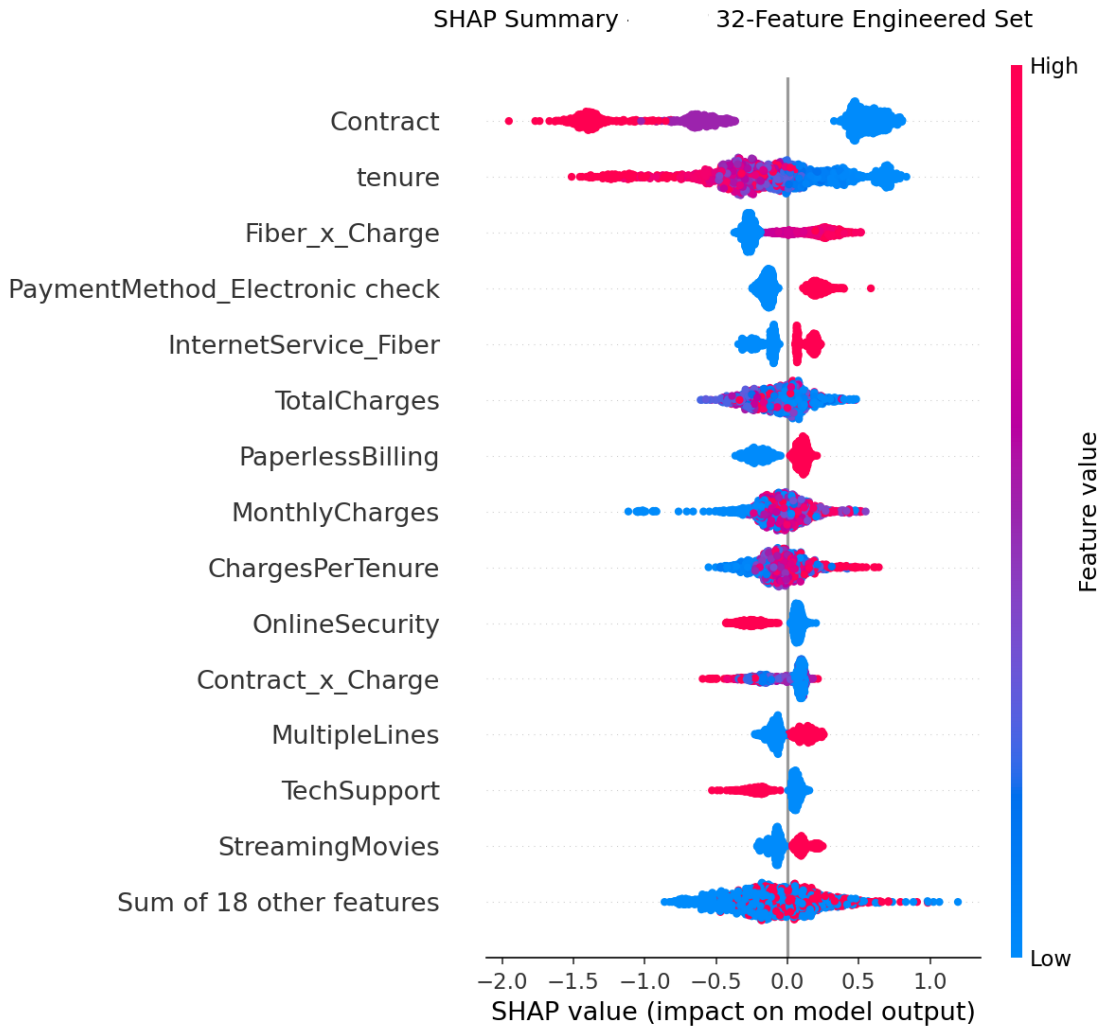


Figure 8: SHAP summary for the 32-feature engineered set. The local image identifies the displayed explainer as XGBoost; it is model-behaviour evidence, not intervention causality.

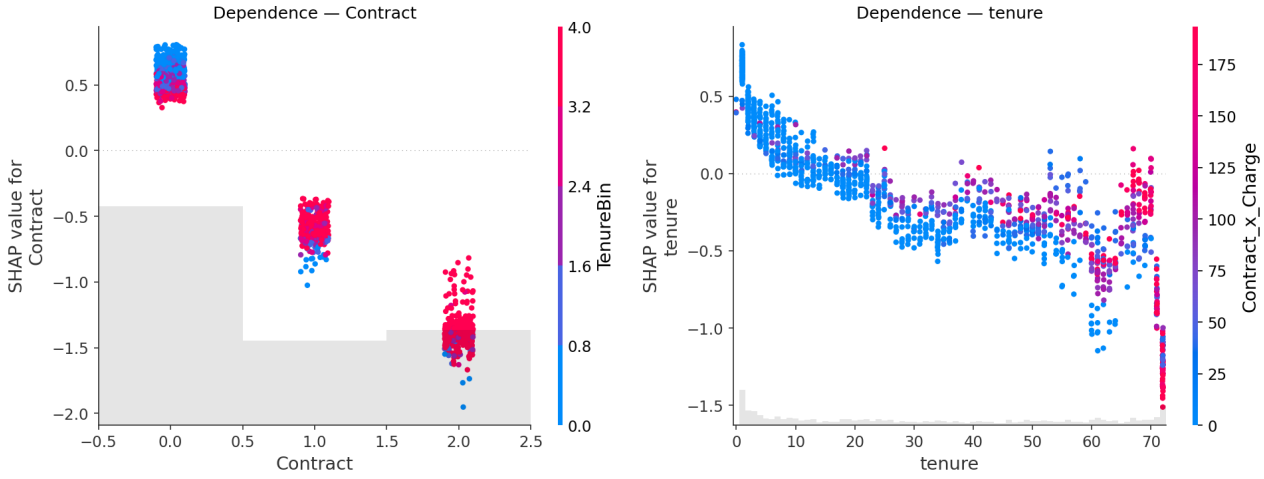


Figure 9: SHAP dependence plots for Contract and tenure in the 32-feature set, using the feature names and axes shown in the local image.

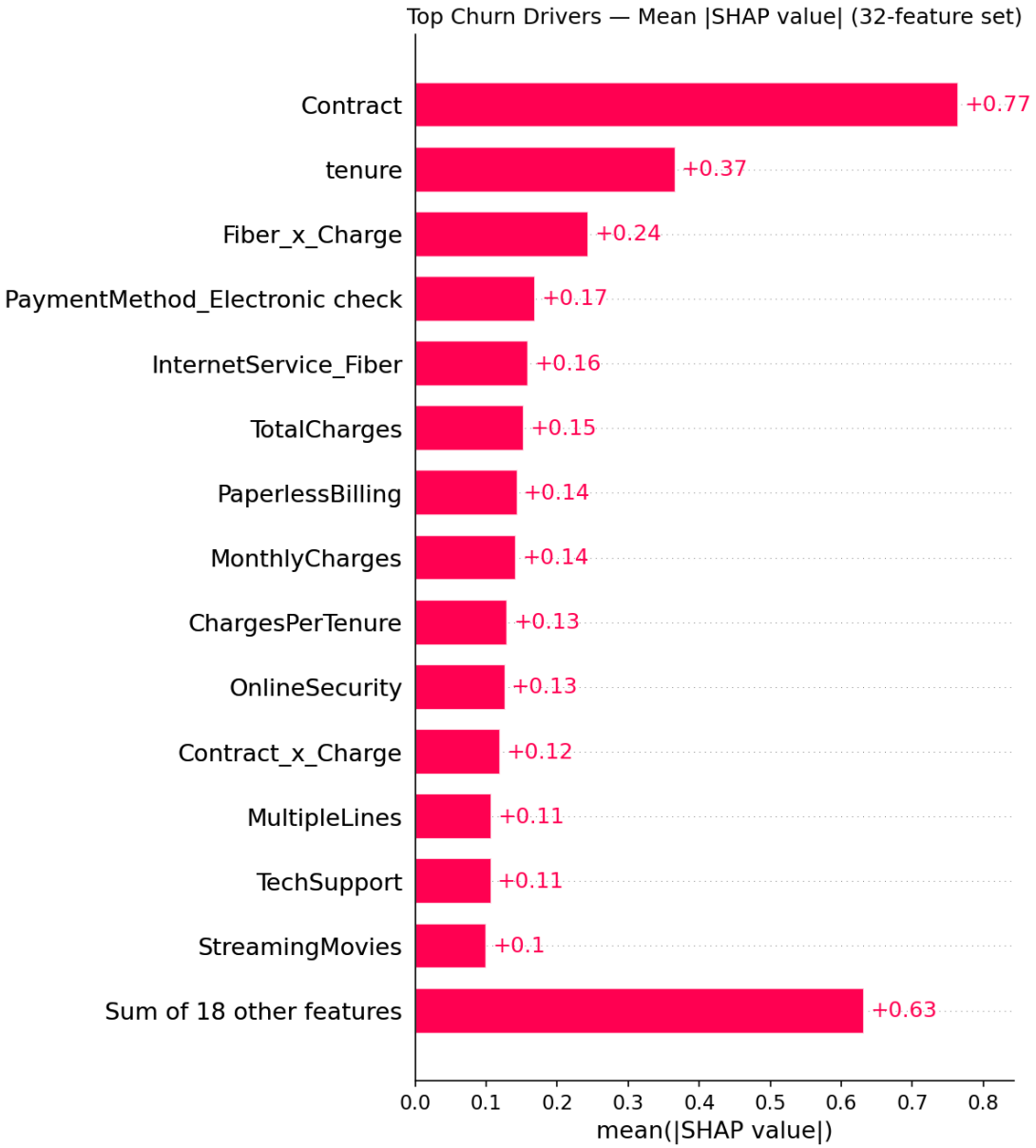


Figure 10: Top churn drivers by mean absolute SHAP value for the 32-feature set.

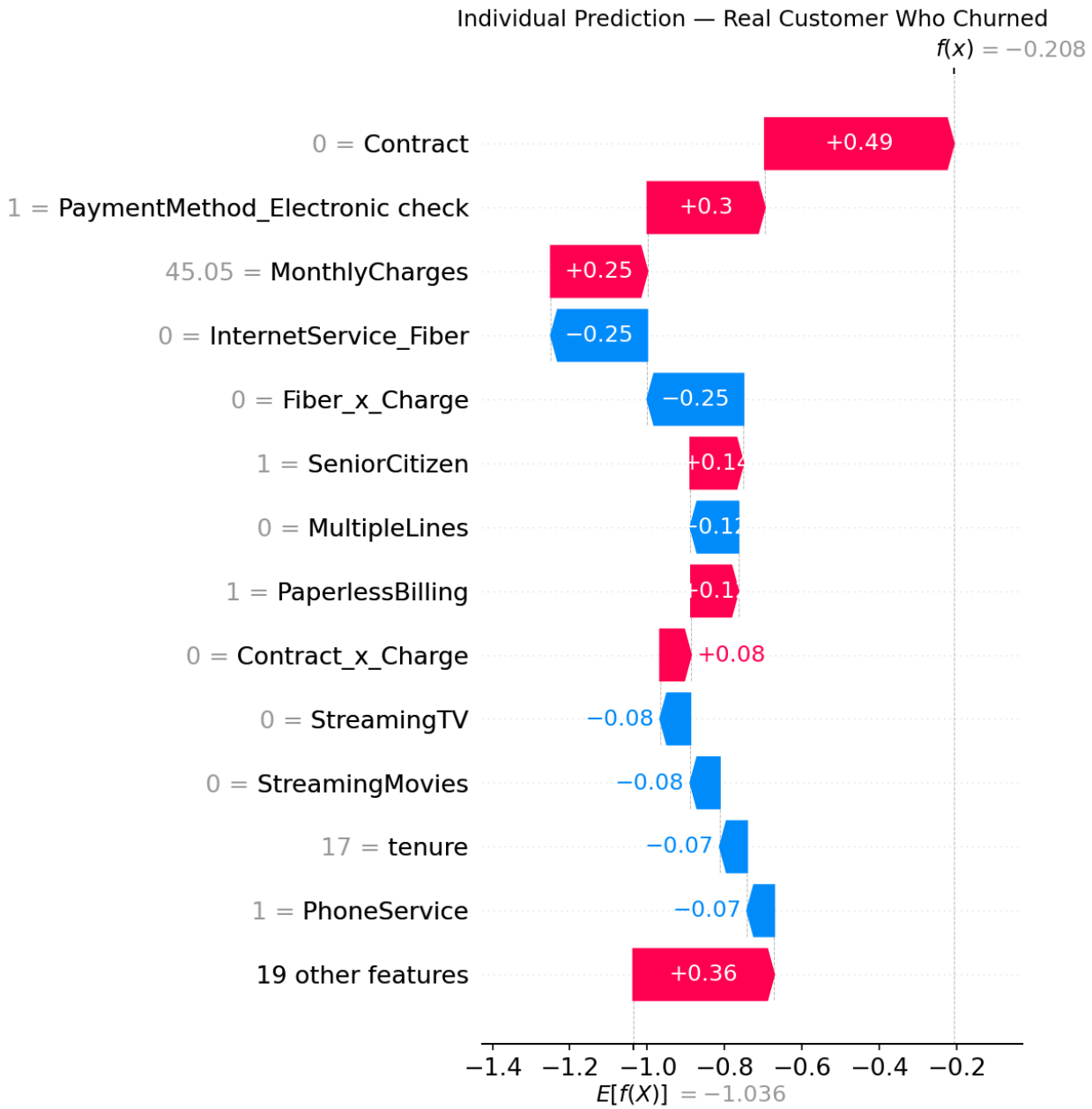


Figure 11: Individual prediction explanation for a real customer who churned, using the 32-feature set. Positive and negative contributions are shown exactly as in the local waterfall image.

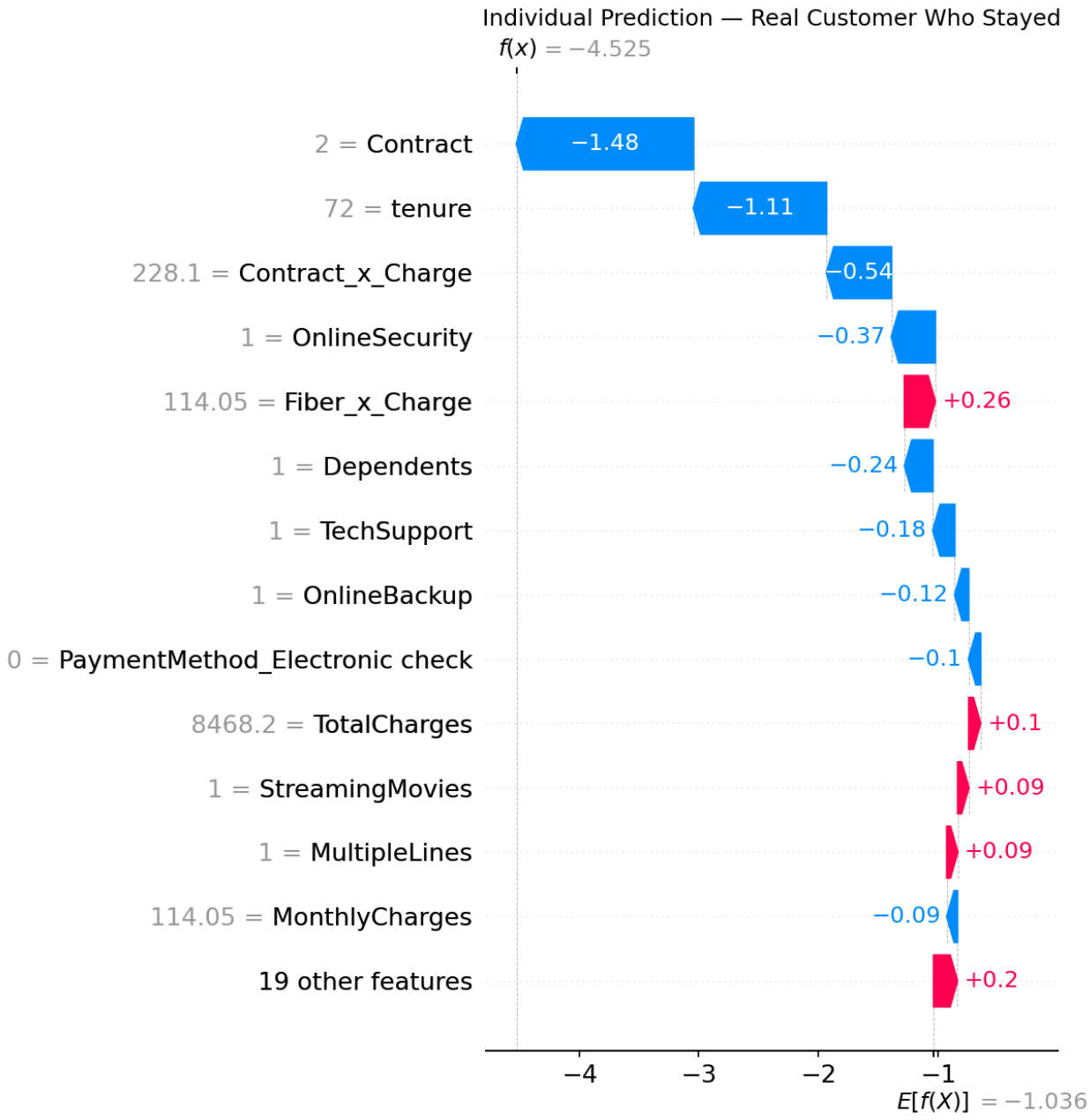


Figure 12: Individual prediction explanation for a retained customer, using the 32-feature set.

6. Prediction-to-action policy

The model output is a probability, not an automatic offer. TCCP applies an explicit policy: rank customers by predicted churn probability, apply a targeting threshold of 0.30 for the retention workflow, and select the highest-ranked customers until the budget is consumed. The conventional 0.50 classification threshold remains visible for comparison but is not assumed to be the economically optimal intervention threshold.

For the demonstration scenario, the save rate is $r = 0.25$, customer value is $v = \$200$, and contact cost is $c = \$5$. The break-even probability implied by the simple utility expression is $c/(rv) = 0.10$; the practical campaign threshold is stricter because contacts are ranked, capacity is limited, and the save-rate assumption is uncertain.

Tier	Risk rule	Suggested action
Immediate intervention	$p \geq 0.50$	High-touch retention offer or priority call
Nurture sequence	$0.30 \leq p < 0.50$	Targeted message, service review, or incentive test
Monitor	$p < 0.30$	Standard service and drift monitoring

Table 3: Demonstration action tiers. The tiers are policy choices and should be validated with intervention outcomes.

At a 20% contact budget, the test-set simulation targets 281 customers and estimates approximately \$7,654 in net value under the stated assumptions. This is a scenario estimate, not an observed causal return. Production use should replace the assumed save rate and value with measured treatment outcomes and confidence intervals.

7. Production and MLOps design

The TCCP production path has four controls. First, the feature contract ensures that inference receives the same 32 engineered columns used during training. Second, the model artifact stores the fitted preprocessing and estimator together, avoiding inconsistent transformations between training and serving. Third, the model registry separates versions and stages; the local registry uses a production stage and the deployment design supports a champion alias. Fourth, operational interfaces expose both batch and online scoring.

The FastAPI service provides health and metadata endpoints and a bounded `/predict` request containing one to 10,000 records. Batch scoring processes input in chunks and appends `predicted_churn_probability` and `predicted_churn`. The CI/CD workflow validates the repository and can run the Databricks job with controlled dependencies. MLflow records parameters, metrics, signatures, and the serialized model for lineage.

Control	TCCP implementation
Reproducibility	Seed 42, versioned code, feature metadata, benchmark CSV, serialized artifacts.
Data safety	Target and identifier removed before inference; unknown categories handled by the encoder.
Model governance	Registry name/stage, champion model metadata, model version, and health endpoint.
Deployment	Batch scoring CLI, FastAPI service, Docker image, GitHub Actions workflow.
Monitoring	Calibration drift, input-schema errors, score distribution, intervention outcomes, and model performance.

8. Limitations, safeguards, and next steps

The dataset is historical and does not establish future performance or causal treatment effects. The random holdout does not replace a time-based holdout when deployment dates are known. SHAP and partial dependence explain the fitted model but cannot establish that a feature causes churn. The retention simulation depends on assumed save rate, value, and cost; each should be estimated from experiments. Subgroup calibration and error rates should also be checked before operational use, particularly for customers who may be disproportionately affected by automated offers.

The next TCCP release should add four safeguards: a time-based validation set, calibration monitoring by cohort, randomized retention experiments, and an approval gate that prevents a model from becoming the production champion without passing data-quality and fairness checks. These controls convert a strong offline benchmark into a responsible decision system.

9. Conclusion

TCCP solves the complete churn workflow rather than optimizing one isolated metric. It starts with a reproducible data contract, compares models under a leakage-safe protocol, selects and evaluates a calibrated champion, explains risk with SHAP and partial dependence, translates probabilities into a budget-aware action policy, and packages the result for governed serving. The central result is not simply an ROC-AUC value: it is a traceable path from a customer record to a defensible retention decision.

Summary for decision-makers

In summary, the system provides three practical outputs: a ranked list of customers at higher risk of churn, a probability that can be checked for reliability, and an explanation of the factors associated with each prediction. The reported ROC-AUC of 0.846 means the model generally ranks a customer who churned above a customer who stayed. The PR-AUC of 0.667 shows that the ranking remains useful for finding likely churners despite class imbalance. The Brier score of 0.137 and the reliability curve indicate how trustworthy the percentages are, but these results should still be monitored on new customer data. Final retention decisions should combine the model with campaign capacity, customer-service judgment, and measured experiment results.

References

1. Blastchar. *Telco Customer Churn Dataset*. Kaggle. Available at: https://www.kaggle.com/datasets/blastchar/telco-customer-churn. Accessed August 2026.
2. TCCP project source files. *Data preparation, calibration, model training, and visualisation implementation*. Local project files: `src/preprocessing.py`, `src/train.py`, `src/calibration.py`, `src/decision_visuals.py`, and `notebooks/03_calibration.ipynb`.
3. TCCP project artifacts. *Benchmark summary, production metadata, and generated figures*. Local project files: `artifacts/benchmark_summary.csv`, `artifacts/production_metadata.json`, and `figures/`.

Appendix A. Production feature schema

The production artifact records the following 32-column feature contract. Identifiers and the target label are excluded before inference.

Table 4: Production feature schema and interpretation.

Feature	Interpretation
SeniorCitizen	Senior-citizen indicator.
Partner	Whether the customer has a partner.
Dependents	Whether the customer has dependents.
tenure	Months the customer has remained active.
PhoneService	Whether phone service is subscribed.
MultipleLines	Multiple-line phone-service category.
OnlineSecurity	Online-security service category.
OnlineBackup	Online-backup service category.
DeviceProtection	Device-protection service category.
TechSupport	Technical-support service category.
StreamingTV	Streaming-TV service category.
StreamingMovies	Streaming-movies service category.
Contract	Contract term category.
PaperlessBilling	Paperless-billing indicator.
MonthlyCharges	Recurring monthly charge.
TotalCharges	Total customer charge.
gender_Male	One-hot encoded gender category.
InternetService_Fiber	Fiber internet indicator.
InternetService_None	No-internet-service indicator.
PaymentMethod_Credit card (automatic)	Automatic credit-card payment indicator.
PaymentMethod_Electronic check	Electronic-check payment indicator.
PaymentMethod_Mailed check	Mailed-check payment indicator.
AutoPay	Automatic-payment feature.
NumServices	Count of subscribed services.
TenureBin	Discretized tenure group.
HighChargeFlag	Indicator for relatively high charges.
Contract_x_Charge	Contract-charge interaction.
Fiber_x_Charge	Fiber-charge interaction.
ChargesPerTenure	Charge normalized by tenure.
LogMonthlyCharges	Log-transformed monthly charge.
LogTotalCharges	Log-transformed total charge.
IsNewCustomer	New-customer indicator.

Appendix B. Reproducibility commands

```
env\Scripts\python.exe src\decision_visuals.py
env\Scripts\python.exe -m src.batch_score \
  --input data/telco_clean_32col.csv \
  --output artifacts/scored_customers.csv
```

The first command regenerates the Matplotlib/Seaborn/SHAP figures from the project artifacts. The second command demonstrates batch prediction through the production feature contract.