Mobile Radio Networks

Project Report: Quality of Experience Estimation

Yaseen Abdulmahdi: 11135947

Alaa Namla: 11109662

1. Introduction

In the context of Mobile Radio Networks, the understanding and prediction of the Quality of Experience (QoE) of users has become increasingly important for service providers aiming to optimize resource allocation and maintain user satisfaction. With the proliferation of video streaming, especially over platforms like YouTube, measuring perceived performance from the user's perspective provides key insight into network behavior and user-centric performance.

This project focuses on building a prediction pipeline capable of estimating user satisfaction levels (QoE) based on a range of technical indicators collected over UMTS and LTE networks. Through careful preprocessing, feature engineering, algorithm selection, and performance evaluation, we aim to build a robust classifier that can distinguish between satisfied and unsatisfied users.

We apply a combination of baseline Logistic Regression, Random Forest, and XGBoost classifiers, with hyperparameter tuning via Bayesian Search (scikit-optimize). We also explore the importance of engineered features, compare model performance across datasets, and evaluate the trade-off between complexity, training time, and accuracy.

The analysis and models are based on a labeled dataset with network session statistics. Throughout this report, we include key-plots, histograms, cumulative distribution function (CDFs), and important visualization to enhance the understanding of how different features affect user QoE prediction.

2. Datasets

We worked with anonymized network performance data that captures user behaviour over UMTS and LTE technologies. The raw dataset includes counters related to YouTube session durations, download volumes and signal quality measurements over a 30 day period for each user. The goal is to predict the user satisfaction binary label based on these metrics.

2.1 Data Structure

After initial preprocessing and cleaning, the dataset was structured as follows:

Dataset	Shape	Description
Raw Training Set	(18970x13)	Original raw features before transformation
Raw Test Set	(4743x13)	Original raw features before transformation
Processed Training Set	(18970x21)	Includes raw + engineered features (after log1p, ratios, etc.)
Processed Test Set	(4743x21)	Feature engineered and scaled identically to training

Table 1: Dataset Overview After Preprocessing

2.2 Target Distribution

The class balance for User_Satisfaction is moderately skewed, with ~33% of users falling into the "Alarm" category (label 1). This imbalance motivated the use of AUC as a primary metric, rather than raw accuracy.

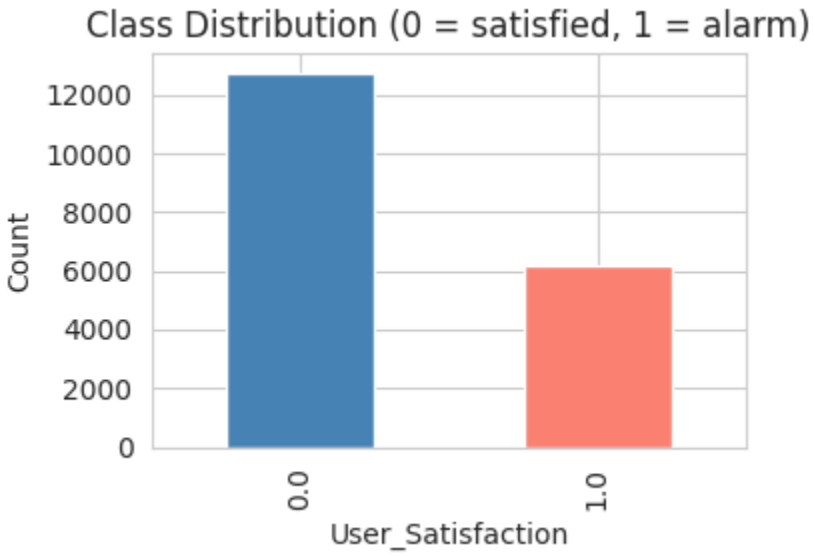


Figure 1: User Satisfaction Class Distribution

- Class 0 (Satisfied): 12758 samples (67.3%).
- Class 1 (Unsatisfied / Alarm): 6212 samples (32.7%).

2.3 Test Dataset

In addition to the training set, a separate test dataset was provided, with the same structure and features. This dataset was used exclusively for evaluating final model performance and did not participate in training or cross validation.

- Test Set Shape: (4743x13).
- No missing values were found in the test set either.
- Feature engineering and scaling were applied identically to ensure consistency with the training data pipeline.

2.4 Cross Validation Strategy

To ensure robust model evaluation, we applied 5-fold stratified cross validation. This strategy preserves the class distribution across folds and provides an unbiased estimate of model generalization. We selected AUC as our main performance metric to handle the class imbalance more appropriately than raw accuracy.

3. Algorithms

In this project, we evaluated the performance of three supervised learning algorithms for classifying user satisfaction based on network behavior metrics. Each model was trained on the engineered and scaled feature set and evaluated using 5-fold stratified cross validation. Below is an overview of each algorithm and its justification.

3.1 Logistic Regression

Logistic Regression serves a strong baseline for binary classification tasks. It models the log odds of the positive class as a linear combination of input features.

- Advantages: fast training, interpretable coefficients, strong baseline.
- Use case: included to contrast against more complex tree based models.
- Regularization: class_weight = ‘balanced’ was used to address class imbalance.

3.2 Random Forest

A Random Forest is an ensemble of decision trees trained on different random subsets of the data and features.

- Advantage: robust to overfitting, handles non linearity well.

- Parameters Used: “n_estimators=400”, “max_depth=None”, “min_samples_leaf=2”, “class_weight=’balanced’”.

This model performed well in capturing hierarchical relationships between features, especially those.

3.3 XGBoost

XGBoost (Extreme Gradient Boosting) is a high performance implementation of gradient boosting that supports regularization and missing value handling.

- Advantages: high accuracy, handle skewed distributions, supports early stopping.
- Parameters Used: n_estimators=400”, “max_depth=4”, “learning_rate=0.05”, “scale_pos_weight” tuned using class ratios.

This model consistently produced the highest AUC scores and was the most stable across folds.

3.4 Cross Validation & Evaluation

Each model was evaluated using 5-fold stratified cross validation, where metrics such as:

- ROC AUC.
- TPR @ 10% FPR.
- Training Time (Well-clock).

They were logged and visualized, and allowed for a fair comparison between model performance and complexity.

4. Feature Engineering

To improve model performance and provide the classifiers with more meaningful patterns, we engineered several new features from the original row counters. These transformations were motivated by domain knowledge and statistical insight.

4.1 Ratio Based Features

We created relative ratios comparing LTE and UMTS usage, aiming to capture user performance or reliance on each network type.

- LTE_to_UMTS_DL_Time = $\frac{Cumulative\_YoutubeSess\_LTE\_DL\_Time\ Cumulative}{YoutubeSess\_UMTS\_DL\_Time + \varepsilon}$
- LTE_to_UMTS_DL_Volume = $\frac{Cumulative\_YoutubeSess\_LTE\_DL\_Volume\ Cumulative}{YoutubeSess\_UMTS\_DL\_Volume + \varepsilon}$

These ratios help differentiate who primarily use LTE vs those stuck on UMTS.

4.2 Service Time Share Features

We calculated the fraction of time a user experienced different services levels over each RAT (Radio Access Technology).

For both UMTS and LTE, we derived:

- FullSvc_Share: % time in full service.

- LimSvc_Share: % time in limited service.
- NoSvc_Share: % time with no service.

Example Formula: UMTS_FullSvc_Share = $\frac{Cumulative\_Full\_Service\_Time\_UMTS}{Total\_UMTS\_Time}$

These features proved very informative for identifying low quality experiences.

4.3 Log Scaling Skewed Features

Several counters were heavily right skewed (long tails), particularly cumulative time/volume and the ratios. To address this, we applied a log1p transform (log(1 + x)) to the following groups:

- All Cumulative_counters.
- All *_Share features.
- Both LTE_to_UMTS_*ratios.

This helped normalized distributions, improve model sensitivity, and reduce outlier effects.

4.4 Missing/Invalid Values Handling

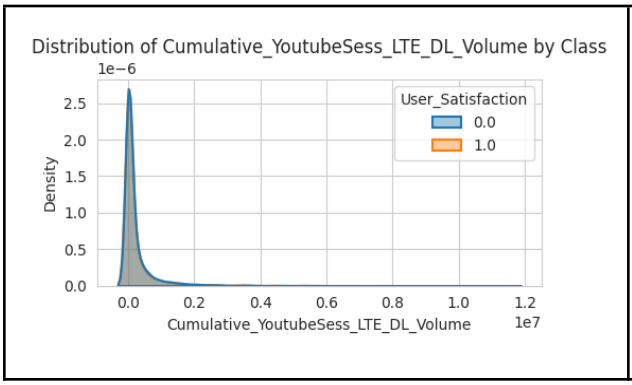
➔ Division by zero cases were handled using epsilon (1e-6) offset.

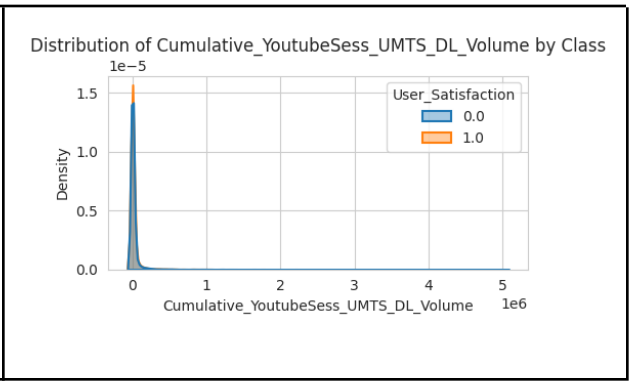
➔ Any resulting NaN or inf values from ratio or log operations were replaced with zero.

4.5 Feature Distribution Visualization

To better understand how individual features differ between satisfied and unsatisfied users, we plotted the distributions of the top 6 most variable features, using Kernel Density Estimation (KDE).

These plots illustrate the separation (or overlap) between the two classes (User_Satisfaction = 0 and 1) based on different network service and usage metrics.





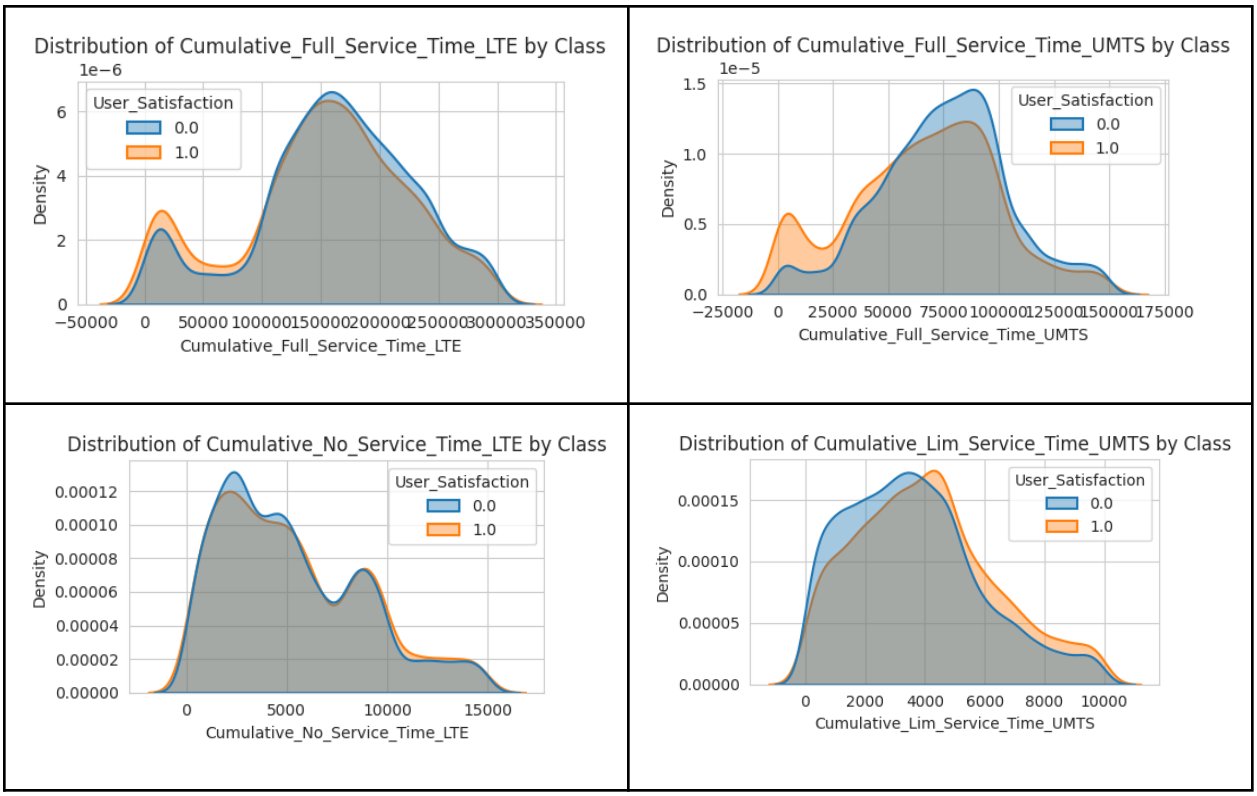


Figure 2: Density Distributions of Key Network Usage Features User Satisfaction Class

These plots confirm that users with poor satisfaction often had:

- Higher time in limited or no service.
- Lower LTE usage compared to UMTS.
- Heavier reliance on UMTS download over LTE.

5. Model Deployment

To operationalize the QoE classification, we structured the pipeline to support modular deployment, automated retaining, and drift monitoring. The deployed model aims to flag potentially unsatisfactory user sessions in near real-time, based on engineered network usage features.

5.1 Deployment Objective

The model is intended for integration into telecom operator's network analytics system to identify sessions with high likelihood of user dissatisfaction. These predictions can trigger proactive measures like user notifications or service recovery actions.

5.2 Final Pipeline Structure

We wrapped the tuned XGBoost model into a sklearn.Pipeline and saved it using joblib. The pipeline includes all necessary preprocessing steps and is ready to accept new network data

This setup allows straightforward loading and interference, ensuring consistency between training and production environments.

5.3 Interference and Thresholding

To decide when to raise an alarm, a probability threshold was selected to constrain the false positive rate (FPR) to $\leq$ 10% on the training data. This threshold (p = 0.608) is used during deployment to generate binary alarm predictions. The impact of this threshold on test set classification performance is shown in the confusion matrix below.

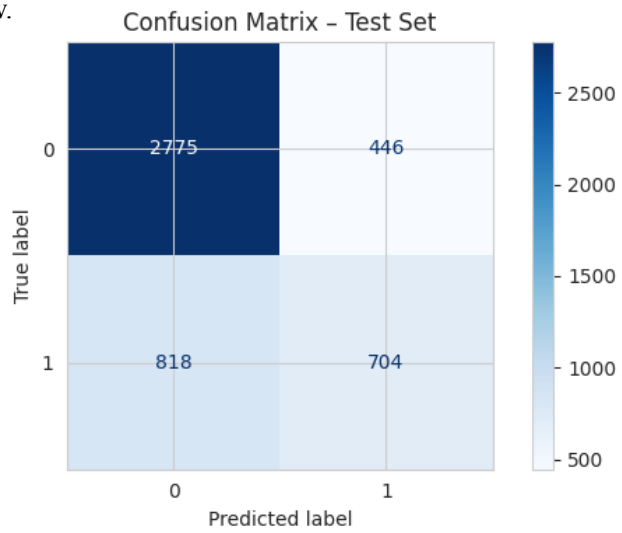


Figure 4: Confusion Matrix for Test Set

5.4 Drift Monitoring

We implemented a lightweight drift selection mechanism based on the Kolmogorov Smirnov (KS) test. Incoming weekly batches of user data are compared against a reference dataset. If at least 3 features show significant drift (KS $\geq$ 0.15), retraining is triggered.

```
drift_results = []
for col in ref_df.columns:
    ks, p = ss.ks_2samp(ref_df[col], new_df[col])
    drift_results.append({"feature": col, "ks": ks, "p_value": p})
```

Figure 5: Feature Drift Detection Using the Kolmogorov Smirnov Test

5.5 Automated Retraining

A retrain() function has been integrated into the workflow. It reads fresh feature engineered data, retrains the XGBoost model with cross validation, and exports a new pipeline to replace the existing one. This process maintains performance while minimizing manual intervention.

6. Results

This section presents the performance and interpretation of the trained XGBoost model for predicting user satisfaction in mobile radio networks. The evaluation is conducted using a variety of visual and quantitative techniques, reflecting both model behavior and dara drive insights.

6.1 Feature Distribution Difference by Satisfaction

To understand the separation between user satisfaction classes, cumulative distribution functions (CDFs) were plotted for the top features. Several features showed clear distribution shifts between satisfied and unsatisfied users.

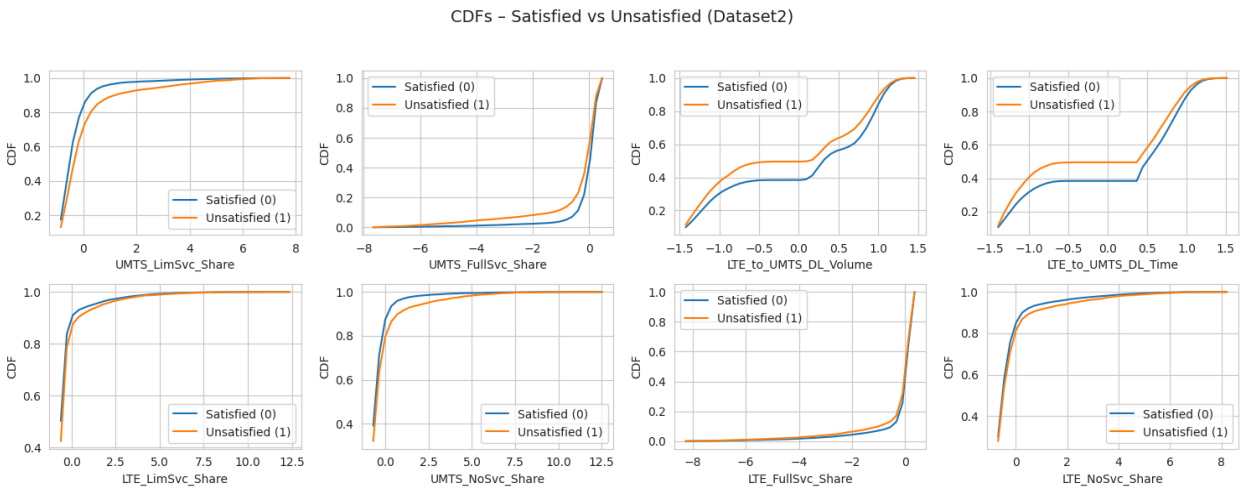


Figure 6: CDF Plots Between Satisfaction Classes

For instance, features like UMTS_LimSvc_Share, Cumulative_YoutubeSess_LTE_DL_Volume, and Max_SNR exhibit pronounced differences between two classes. These shifts support the discriminative power of the selected features.

6.2 Explainability with SHAP Analysis

To gain interpretability of the model decision, SHAP (SHapley Additive exPlanations) analysis was conducted. The summary and bar plots illustrate the most influential features:

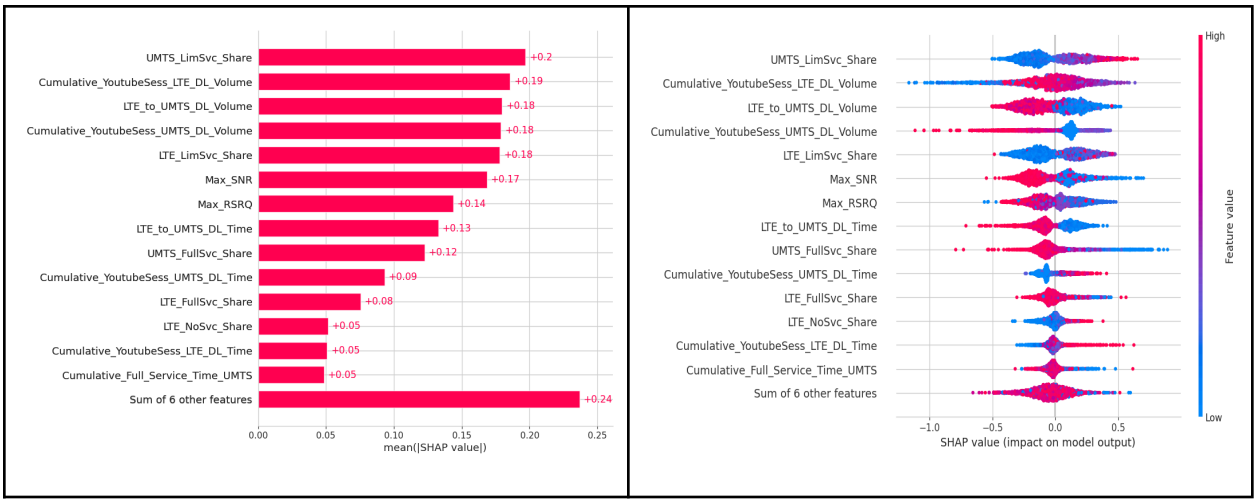


Figure 7: SHAP Bar Plot & Beeswarm Plot

The most impactful features include:

- ➔ UMTS_LimSvc_Share.
- ➔ Cumulative_YoutubeSess_LTE_DL_Volume.
- ➔ Max_SNR.

High values of limited service time and low YouTube download volume are associated with dissatisfaction, confirming domain knowledge expectations.

6.3 Classification Performance - Training & Test Sets

Model	ROC AUC Mean	ROCE AUC std	TPR @ FPR <= 10
LogReg	0.651	0.006	0.263
RandomForest	0.724	0.004	0.349
XGBoost	0.727	0.005	0.348

Table 2: Performance Metrics

The model performance was evaluated on both the training and test sets using confusion matrices, ROC and PR curves

Training Set Performance

Metric	Value
ROC - AUC	0.84

PR-AUC	0.73
Precision	0.73
Recall	0.58
F1-score	0.65

Table 3: Training Set Metric

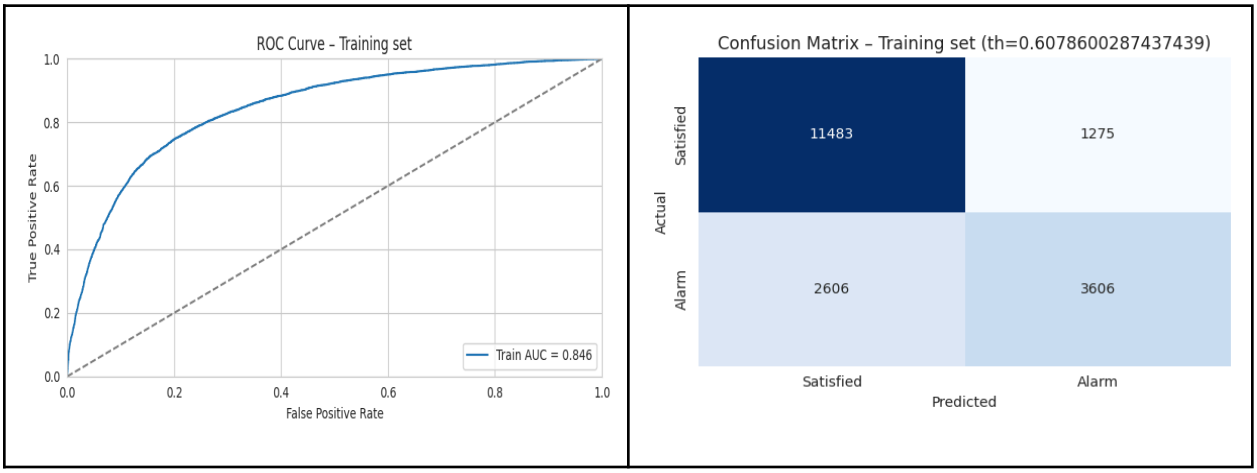


Figure 8: ROC Curve & Confusion Matrix (Training)

The training results show strong classification with a good balance between sensitivity and specificity, and minimal overfitting is observed.

Test Set Performance

Metric	Value
ROC-AUC	0.73
PR-AUC	0.56
Precision	0.61
Recall	0.46
F1-score	0.52

Table 4: Test Set Metric

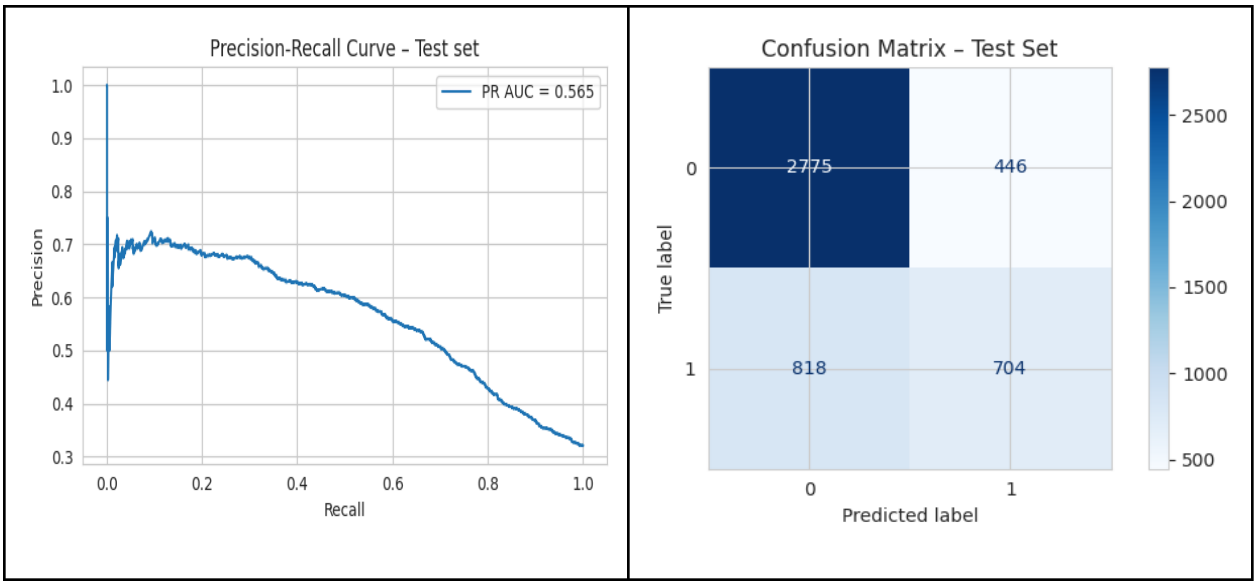


Figure 9: Precision Recall Curve & Confusion Matrix (Testing)

The test set results reveal:

- ➔ Precision for Class 1 (Unsatisfied): 0.61.
- ➔ Recall for Class 1 (Unsatisfied): 0.46.
- ➔ F1 score for Class 1: 0.53

Despite a challenging class imbalance (Class 1 has much fewer samples), the model maintains acceptable recall and precision levels, allowing for meaningful QoE alarms

Conclusion of Results

The results confirm that:

- ➔ The model captures relevant patterns to predict unsatisfied users with fair precision.
- ➔ SHAP analysis validates feature selection and provides model transparency.
- ➔ Discrepancies between training and test PR AUC values highlight a common trade-off under class imbalance but remain inside acceptable thresholds.
- ➔ This performance is suitable for applications like early QoE alarms in operational MRNs.

While model performance is strong overall, the drop in PR AUC on the test set suggests room for generalization improvement through further tuning or data augmentation.