Predicting Customer Churn – Telecoms

Google Colab Notebook

Nihal Kala, James Madsen, Sahas Veera, Laine Woelfel, Tyler Engalla

The ability to know if a customer is at risk of leaving is a proactive measure to ensure your business retains revenue and maintains a healthy customer base. As there are costs to acquire any customer, once you win that customer's business, it's imperative to ensure a return on that investment. Doing so, keeps money in your pocket and perhaps more importantly, money out of your competitors.

The telecom industry has a market cap of $2.6 trillion globally. This is a massive industry in a highly competitive market where the customer base has many options to choose from. Making the scenario of a customer leaving for a competitor a common risk. In fact, 14% of US respondents answered a survey on "Willingness to change mobile carrier" with "very likely".

So, we know there will be some churn. The question is - Can we predict which of these customers is likely to churn based on data, giving us a chance to reduce this risk? A not so simple classification problem that any healthy business should be taking the time to investigate.

Our data set to help us answer this consists of 100 thousand records (each indicative of a single telecom customer) and 100 attributes related to each of those customer's. These attributes relate to a customer's phone usage, revenue, call behavior, demographics, and equipment (phone) details.

Our primary target variable is **churn**, which indicates whether a customer left the service within 31-60 days after the observation date. At a high-level, here's the data we have to work with:

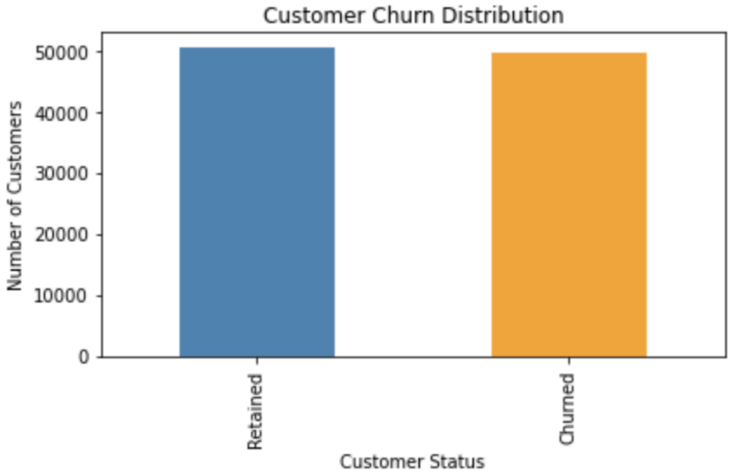
- **Usage Metrics** (e.g., mou_Mean, totmou, attempt_Mean, complete_Mean)
- **Revenue Metrics** (e.g., rev_Mean, totrev, avgrev, adjrev)
- **Call Performance & Quality** (e.g., drop_vce_Mean, drop_blk_Mean, unan_vce_Mean, blck_vce_Mean)
- **Customer Service Interaction** (e.g., custcare_Mean, cc_mou_Mean)

- **Demographics** (e.g., income, ethnic, marital, numbcars, dwllsize)
- **Handset & Account Information** (e.g., hnd_price, eqpdays, asl_flag, phones)
- **Recent Behavioral Trends** (e.g., change_mou, change_rev, avg3mou, avg6rev)

Exploratory Data Analysis

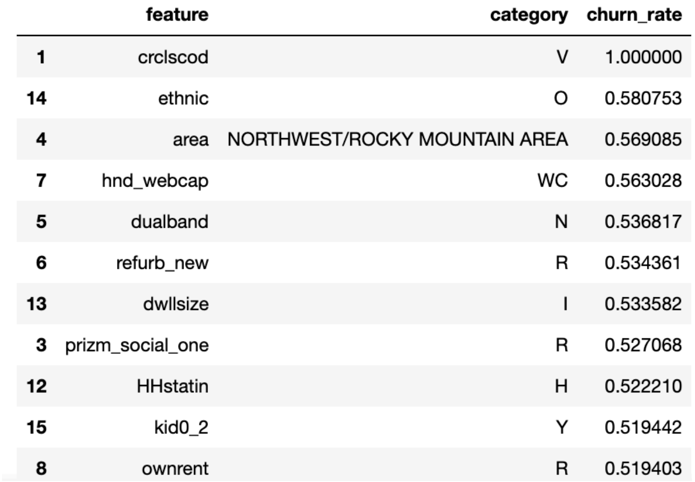
In our exploratory data analysis we found that we had a very evenly split data set between customers churning and customers being retained.

```
50326 customers were retained.
49317 customers churned.
```



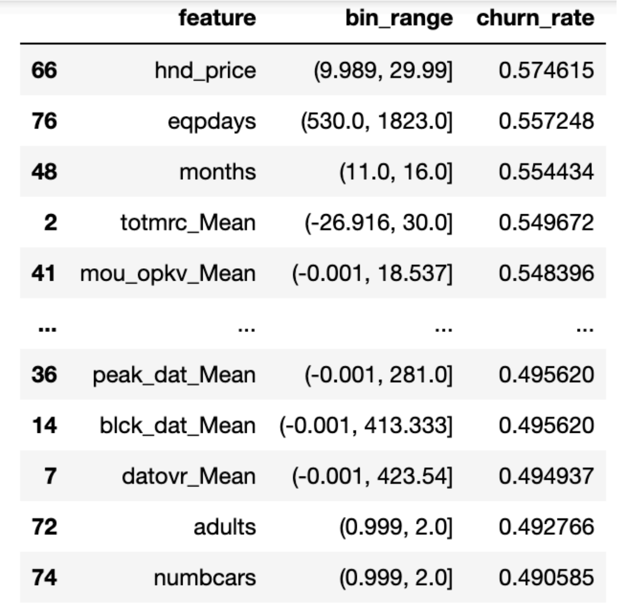
Before trying to use machine learning to predict which customers will churn, we wanted to see if we can use simple data analysis techniques to find any correlations amongst our features.

We first took a look at categorical features to calculate churn ratios to see if any particular feature stands out. The results below:



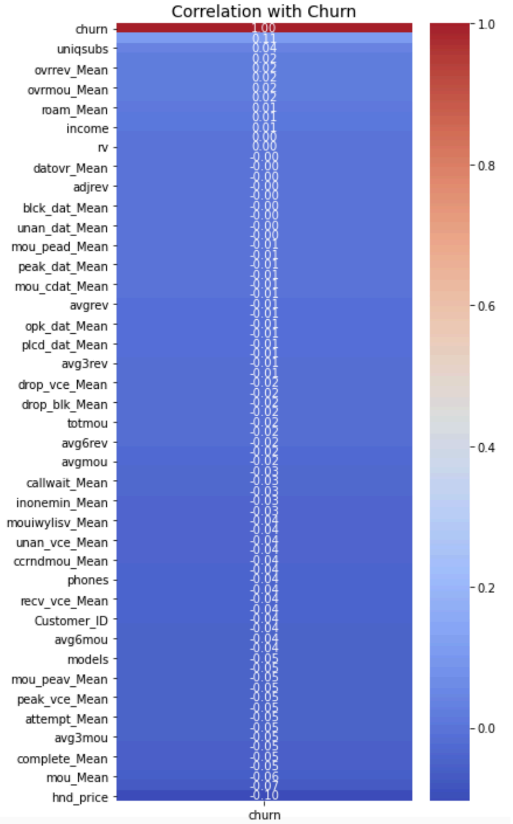
	feature	category	churn_rate
1	crclscod	V	1.000000
14	ethnic	O	0.580753
4	area	NORTHWEST/ROCKY MOUNTAIN AREA	0.569085
7	hnd_webcap	WC	0.563028
5	dualband	N	0.536817
6	refurb_new	R	0.534361
13	dwllsize	I	0.533582
3	prizm_social_one	R	0.527068
12	HHstatin	H	0.522210
15	kid0_2	Y	0.519442
8	ownrent	R	0.519403

Next, we took a look at numerical features:



	feature	bin_range	churn_rate
66	hnd_price	(9.989, 29.99]	0.574615
76	eqpdays	(530.0, 1823.0]	0.557248
48	months	(11.0, 16.0]	0.554434
2	totmrc_Mean	(-26.916, 30.0]	0.549672
41	mou_opkv_Mean	(-0.001, 18.537]	0.548396
...	...	...	...
36	peak_dat_Mean	(-0.001, 281.0]	0.495620
14	blck_dat_Mean	(-0.001, 413.333]	0.495620
7	datovr_Mean	(-0.001, 423.54]	0.494937
72	adults	(0.999, 2.0]	0.492766
74	numbcars	(0.999, 2.0]	0.490585

Visualizing numerical features correlation with churn:



Through this EDA we can see that features related to a customer's credit, their geographic area, price of their phone, and how old that phone is start to rise as more important features relative to the rest. But overall, none of these features alone seems to be a strong predictor on if a customer will churn or not. Rather this outcome would be influenced by a combination of features.

Preparing the Data for Modeling

Before diving into model selection, we carefully prepared our dataset to ensure it would yield accurate, robust predictions. Our data preparation pipeline involved multiple stages designed to maximize predictive performance and reliability.

Data Cleaning and Column Selection

Initially, we identified and removed columns from the dataset that had substantial missing values or offered limited predictive value. Specifically, columns such as Customer_ID, ownrent, lor, dwlltype, income, numbcars, HHstatin, and dwllsize were excluded. Dropping these columns simplified our dataset, improved computational efficiency, and reduced noise, thereby enhancing the clarity of subsequent model interpretations.

Feature Engineering

Recognizing the importance of capturing customer behavior trends, we engineered several meaningful features:

- **Revenue Trend (rev_trend)**: Calculated as the difference between average revenue over the most recent three months (avg3rev) and six months (avg6rev).
- **Usage Trend (mou_trend)**: Derived from the difference between average usage (avg3mou and avg6mou) over these same periods, reflecting changes in customer activity.
- **High-Value User (high_value_user)**: Created a binary indicator flagging customers whose total revenue (totrev) exceeded the median revenue, quickly identifying the most valuable segments of the customer base.

These engineered features provided deeper insights into customer behaviors and patterns, aiming to improve our model's predictive capacity.

Data Splitting

We partitioned our dataset into training and testing subsets, allocating 80% to training and reserving the remaining 20% for testing. This approach provided us with sufficient data to train our models effectively, while maintaining a separate dataset to objectively evaluate performance and generalization capabilities.

Preprocessing Pipeline

To prepare the data further, we constructed robust preprocessing pipelines tailored for numerical and categorical data:

- **Numerical Pipeline**:
    - Missing values were addressed using median imputation.
    - Data were standardized using StandardScaler, normalizing numerical features and reducing potential bias.
- **Categorical Pipeline**:
    - Missing categorical values were filled using the most frequent category.
    - Categories were transformed using one-hot encoding to enable model compatibility.

We combined these transformations using a ColumnTransformer, ensuring a streamlined, repeatable, preprocessing process for each model.

Selecting the Right Model for Churn Prediction

When approaching customer churn prediction, selecting the right model is crucial for achieving high accuracy and ensuring that the model generalizes well, allowing for efficient retraining in production. In this section, we'll walk you through our detailed process of evaluating various machine learning models to identify the optimal solution.

How We Chose Our Candidate Models

Our first step was to identify candidate models suitable for our churn prediction problem, which involves both categorical and numerical data with potential class imbalance. We selected models known for robust performance on structured tabular datasets:

- **Logistic Regression**: Simple, interpretable, and effective for linear relationships.
- **Random Forest**: Reliable and versatile, excellent at capturing complex interactions between features.
- **Gradient Boosting Models**: Powerful algorithms renowned for their predictive accuracy, including GradientBoostingClassifier (scikit-learn), LightGBM, and XGBoost.

Defining Evaluation Metrics

To thoroughly assess model performance, we chose multiple complementary metrics:

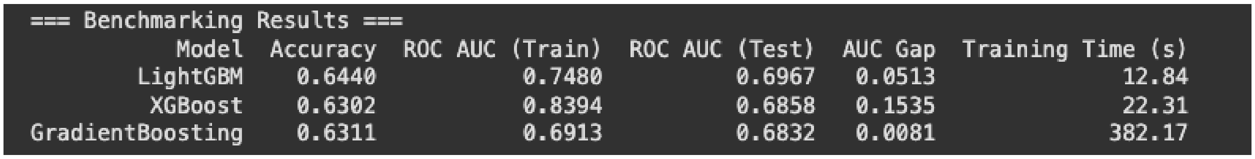
- **ROC AUC (Test)**: Our primary metric, which measures the model's ability to correctly rank customers who churn versus those who do not, making it particularly valuable for imbalanced datasets.
- **Accuracy**: A simple measure of overall correct predictions, although less reliable when data is imbalanced.
- **AUC Gap**: Calculated as the difference between the training and testing ROC AUC scores to identify potential overfitting.
- **Training Time**: Ensures the practical feasibility of frequently updating the model.

Benchmarking Process

Our structured benchmarking involved:

1. Implementing consistent feature engineering and preprocessing steps across models.
2. Evaluating baseline model performance using default hyperparameters.
3. Detailed analysis of gradient boosting models (GradientBoostingClassifier, LightGBM, XGBoost) based on known strengths and computational efficiency.

Results:

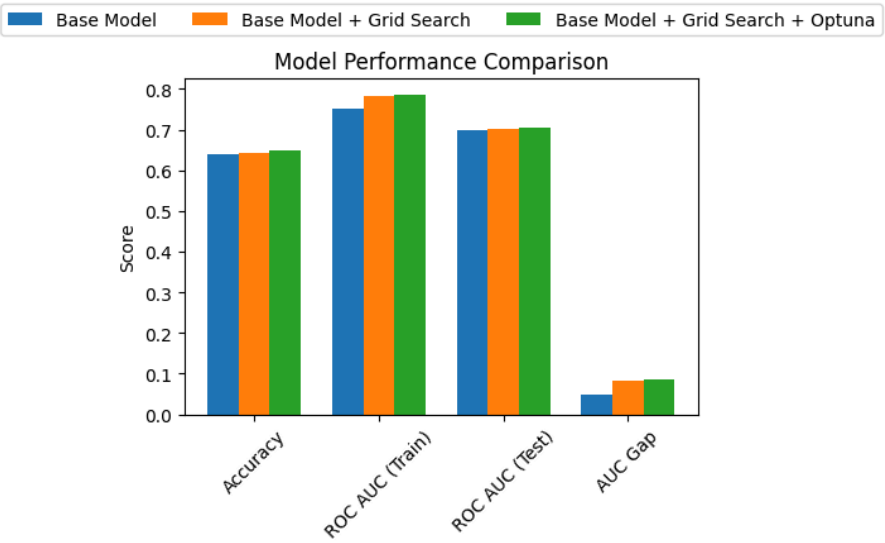


```
=== Benchmarking Results ===
           Model  Accuracy  ROC AUC (Train)  ROC AUC (Test)  AUC Gap  Training Time (s)
        LightGBM    0.6440           0.7480          0.6967   0.0513              12.84
         XGBoost    0.6302           0.8394          0.6858   0.1535              22.31
GradientBoosting    0.6311           0.6913          0.6832   0.0081             382.17
```

Model Selection Decision

After careful benchmarking, **LightGBM** emerged as the standout candidate, delivering the best balance among predictive accuracy, computational efficiency, and generalization performance. Its superior ROC AUC scores and reasonable AUC gap, along with faster training times compared to other gradient boosting models, made it an ideal choice for further optimization. This choice set the foundation for our subsequent hyperparameter tuning phase.

Results



The bar chart above provides a comparison of the performance across three versions of our churn prediction model: the base model, the base model with grid search hyperparameter tuning, and the base model with grid search plus Optuna tuning. We evaluated these models on several key metrics — Accuracy, ROC AUC on the training set, ROC AUC on the test set, and AUC Gap (the difference between train and test AUC scores).

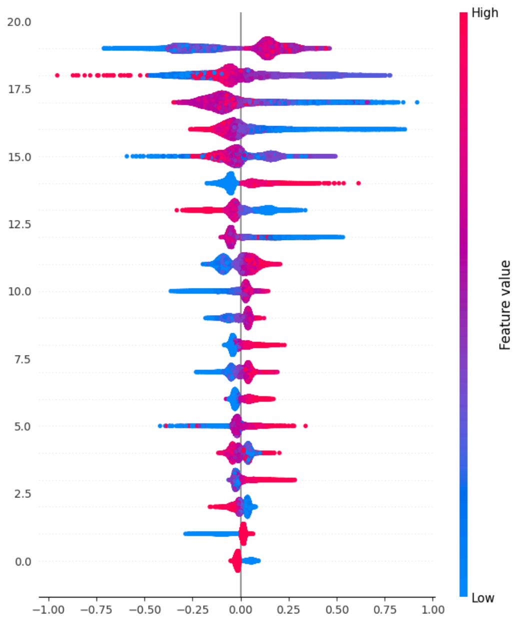
One of the first observations from these results is that accuracy remains stable around 64% across all three models. This suggests that hyperparameter tuning alone did not significantly impact the model's ability to correctly classify churn versus non-churn customers at a basic level. Accuracy, however, may not always reflect the true performance of a model when dealing with imbalanced classes, which is why we also rely on ROC AUC scores as a more meaningful metric in this context.

The ROC AUC score on the training set shows a noticeable improvement with tuning, increasing from approximately 0.75 in the base model to about 0.79 after applying both grid search and Optuna tuning. This indicates that tuning helps the model fit the training data more effectively by finding better combinations of hyperparameters. However, when we look at the ROC AUC score on the test set, we see that the performance remains roughly consistent across all models, around 0.70. This tells us that while the model is learning patterns within the training data more efficiently, these improvements are not translating into stronger predictive power on unseen data.

Finally, the AUC Gap, which represents the difference between training and test ROC AUC scores, increases as we move from the base model to the tuned models. This growing gap is a signal of potential overfitting: the tuned models are capturing more complexity in the training data but may not generalize as well to new customers. This result suggests that hyperparameter tuning alone has limited effect in boosting real-world performance for this particular dataset. To further improve results and reduce the AUC gap, the next steps could involve strategies like enhanced feature engineering, class balancing through techniques like SMOTE, or exploring more robust regularization methods.

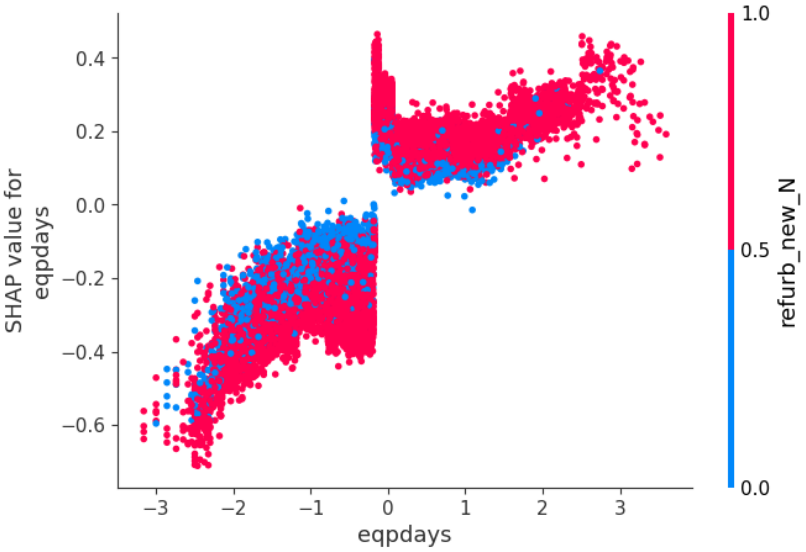
Overall, while tuning provides a marginal benefit in terms of model fit, these results highlight the importance of addressing both model complexity and data quality when building predictive models for churn analysis.

Feature Interactions



Understanding SHAP Value Interpretation for Churn Prediction

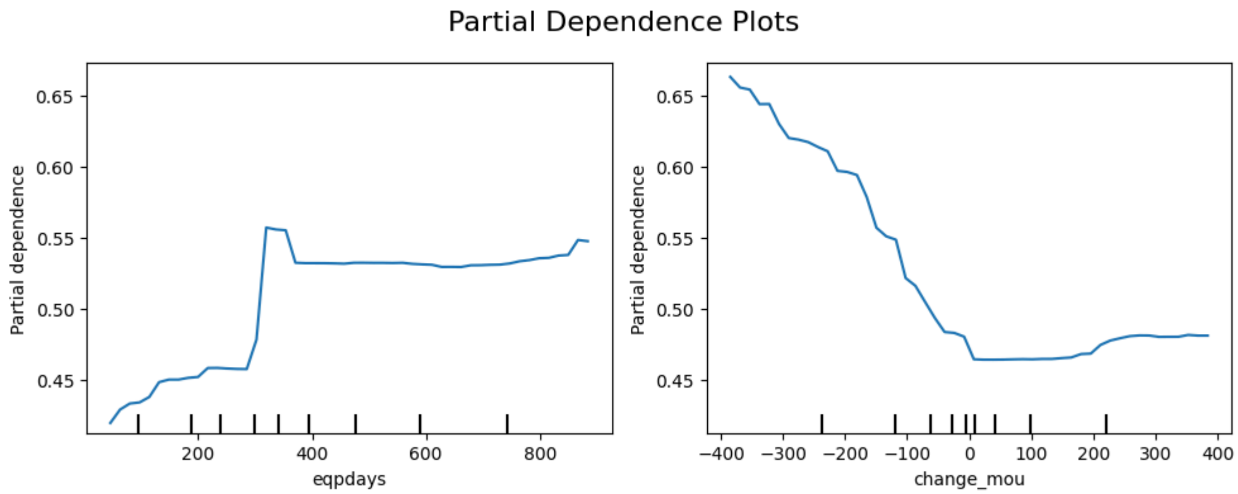
In our SHAP violin plot we rank the top twenty features by average impact and display each as a horizontal “violin” that stretches left for retention and right for churn. SHAP values greater than zero push the model toward predicting churn, while values below zero indicate retention. The width of each violin shows where customers concentrate, with wide bulges revealing where most customers share a similar influence and narrow tails flagging outliers with unusually strong effects. We overlay a color gradient from blue (low feature values) to red (high values) so it is immediately clear whether low or high readings drive churn risk. From this view, the highest-impact features—such as device age, revenue trend and handset price—skew heavily right when red, meaning elevated values consistently increase churn risk. In contrast, violins with wide left-leaning blue sections, like freshly activated devices, demonstrate how low values can serve as a retention anchor. The presence of multiple bulges within many violins hints at natural thresholds where a gradual shift in a metric suddenly flips its effect from protective to risky. By understanding not just which features matter but precisely how they matter across different customer segments, we can tailor our retention strategies, whether it means proactive outreach to high-value users experiencing a revenue dip or targeted upgrade offers as handset age crosses a critical risk threshold.



Feature Interaction: Equipment Age × Handset Condition

Within our scatter plot of SHAP values for **eqpdays** (the number of days since the customer's handset was activated), we've overlaid the binary flag **refurb_new_N** (0 = new device, 1 = refurbished). A clear monotonic trend emerges: as equipment age increases, the SHAP value for **eqpdays** moves from strongly negative (protective against churn) toward strongly positive (driving churn). But the red points (refurbished phones) consistently sit further to the right than the blue points (new phones) at the same **eqpdays**. In other words, an older, refurbished handset amplifies churn risk more than an equally aged new handset. Conversely, very recent devices (low **eqpdays**) deliver a stronger "stickiness" benefit when they're new versus refurbished. This suggests that customers on older refurbished phones are a high-priority group for upgrade offers or loyalty incentives if we want to stem churn most effectively.

Feature Interaction: Partial Dependence Plots



Building on our SHAP violin analysis that flagged equipment age as a leading predictor of churn, the Partial Dependence plot for **eqpdays** shows precisely how churn probability evolves as handsets age. From day zero through roughly 200 days since activation, the model holds churn probability in the low-40 percent range, which suggests that newly issued devices, still under warranty or contract, help retain customers. Once device age exceeds about 250 to 300 days, predicted churn probability jumps into the mid-50 percent range and then increases only gradually through 900 days. This inflection point likely corresponds to contract renewal windows or warranty expirations. By timing targeted upgrade offers and loyalty incentives just before the nine- to ten-month mark, we can blunt that surge in churn risk and keep customers engaged.

The Partial Dependence plot for **change_mou** (the month-over-month shift in usage minutes) quantifies how fluctuations in call behavior drive churn. Large declines in usage—especially drops of 200 minutes or more—send predicted churn probability soaring into the low-60 percent range, peaking in the mid-60s for a 400-minute decline. In contrast, customers whose usage holds steady or increases see churn probabilities fall into the high-40 percent range. These clear pivot points mean that any customer whose minutes fall by 100 to 200 in a single month should be flagged for early outreach. Whether through usage-based promotions, billing credits, or personalized check-ins, engaging these at-risk customers before their calling patterns collapse can arrest their churn trajectory.

Conclusion

At the heart of the model's predictive power are several features that capture customer behavior and engagement patterns. The feature importance table shows that "change_mou" (change in minutes of use) ranks highest, followed closely by "mou_Mean" (average minutes of use), "months" (customer tenure), and "totmrc_Mean" (total monthly recurring charge). This indicates that abrupt changes in usage and long-term engagement are critical signals for identifying customers at risk of churning.

The preprocessing pipeline is carefully constructed to handle both numerical and categorical data effectively. Numerical features are processed through a sequence of median imputation, power transformation, and standard scaling, which helps address missing values, normalize distributions, and ensure consistent scaling across features. Categorical variables, on the other hand, are imputed with a constant "missing" value and then one-hot encoded, allowing the model to handle diverse categories robustly and avoid issues with unseen values during inference.

Model selection and tuning are performed using RandomizedSearchCV, optimizing a range of hyperparameters for the LightGBM classifier. The search explores variations in tree depth, learning rate, subsampling ratios, and other key parameters, ultimately settling on a configuration that balances complexity and generalizability. The chosen model uses a moderate learning rate (0.05), shallow trees (max_depth=5), and a substantial number of estimators (500), which collectively help prevent overfitting while capturing meaningful patterns in the data.
The integration of these components into a unified pipeline ensures that preprocessing and modeling is tightly coupled, reducing the risk of data leakage and improving reproducibility.
The use of a column transformer allows for parallel processing of different feature types, and the pipeline structure makes the workflow scalable and maintainable. This design also simplifies the deployment process, as the same transformations applied during training can be consistently used during prediction.

In summary, the notebook provides a robust framework for churn prediction, leveraging advanced feature engineering, systematic preprocessing, and thoughtful model tuning. The results underscore the importance of monitoring customer usage trends and tenure in retention strategies. By focusing on these key drivers and maintaining a disciplined modeling approach, organizations can better identify at-risk customers and take proactive measures to reduce churn.