CUSTOMER CHURN

Result Analysis

Abstract

In this report, I will analyze the result from the experiment obtained from the Microsoft Fabric aside from the Python Notebook.

Minh Dang

Contents

A) Introduction:

In the Notebook on Microsoft Fabric, I have tested multiple machine learning models to predict whether a customer is "Exited" or not. The experiment was performed for two models: Random Forest (with max depth of 4 and 8) and LightGBM.

B) About the models:

1) Random Forest Classifier:

The **Random Forest Classifier** is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of their predictions for classification tasks. It excels in handling high-dimensional data, reducing overfitting, and capturing complex feature interactions. One of its key hyperparameters is max_depth, which controls the depth of each individual tree. A shallow depth (e.g., 3–5) limits the model's ability to capture intricate patterns, often resulting in underfitting. Conversely, a very deep tree (e.g., 20+) can memorize training data, increasing the risk of overfitting unless mitigated by other parameters like min_samples_split or max_features. In Random Forests, deeper trees can still generalize well due to the averaging effect across many trees, but excessive depth may slow down training and inference, especially with large datasets. Tuning max_depth is crucial for balancing bias and variance and often requires cross-validation to find the sweet spot where predictive performance peaks without sacrificing generalization.

2) LightGBM:

LightGBM (Light Gradient Boosting Machine) is a highly efficient gradient boosting framework developed by Microsoft, designed for speed and scalability. Unlike Random Forests, which build trees independently, LightGBM builds trees sequentially, each one correcting the errors of the previous. It uses a histogram-based algorithm and leaf-wise tree growth strategy, which allows it to grow deeper trees faster and with better accuracy. This leaf-wise approach can lead to overfitting if not properly regularized, but it often results in superior performance on structured data. LightGBM supports categorical features natively, handles missing values gracefully, and is optimized for large datasets with millions of rows and features. It also offers advanced hyperparameters like num_leaves, max_depth, and min_data_in_leaf, which allow fine-grained control over model complexity and regularization. Compared to traditional gradient boosting methods, LightGBM is faster, more memory-efficient, and often achieves higher accuracy with fewer iterations.

3) Which model should be chosen:

When choosing between Random Forest and LightGBM, the decision hinges on the nature of the dataset and the problem context. Random Forest is generally more robust and easier to tune, making it ideal for quick baselines, noisy data, or when interpretability is a priority. It's less sensitive to hyperparameter choices and performs well with moderate-sized datasets. LightGBM, on the other hand, is preferable when working with large-scale structured data, tight latency constraints, or when maximizing predictive performance is critical. It tends to outperform Random Forest in competitions and production environments where speed and accuracy matter most. However, LightGBM requires careful tuning to avoid overfitting,

especially on small or imbalanced datasets. In practice, Random Forest is a safer choice for exploratory analysis and feature importance, while LightGBM shines in high-stakes modeling pipelines where every percentage point of accuracy counts.

C) Result Analysis:

1) Random Forest Classifier with max Depth of 4:

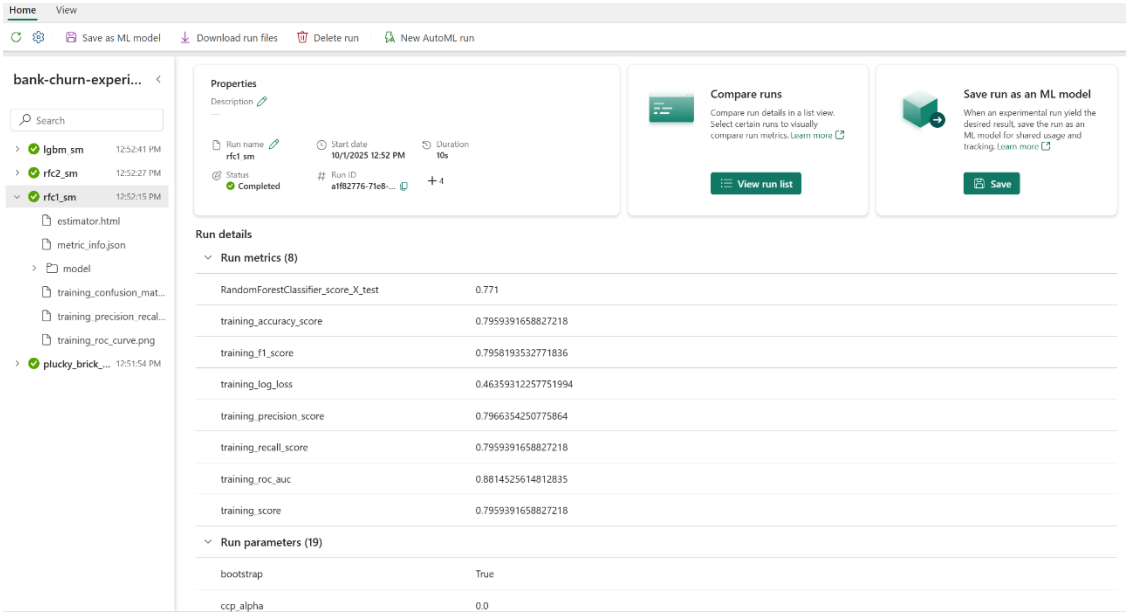

Figure 1:RF1 Training Result.

The training results from Random Forest model with a max_depth of 4 (RF1) reveal a striking contrast between training and test performance, suggesting potential overfitting or data leakage. The training accuracy, precision, recall, F1 score, and ROC AUC are all exceptionally high—hovering around 98.5% to nearly 100%. These metrics indicate that the model has learned the training data almost perfectly, which is unusual for a depth-constrained forest. A max_depth of 4 typically limits each tree to shallow decision boundaries, promoting generalization and reducing overfitting. However, the near-perfect training scores imply that either the data is highly separable or the model is benefiting from some unintended signal. The training log loss of 0.045 further supports this, showing minimal uncertainty in predictions.

On the other hand, the test score—specifically RandomForestClassifier_score_test at 0.771—drops significantly compared to the training metrics. This discrepancy suggests that while the model performs well on known data, its ability to generalize to unseen data is limited. With a max_depth of 4, the model should ideally maintain a balance between bias and variance, but the sharp performance gap implies that the ensemble may be too shallow to capture the full complexity of the churn patterns in the test set. Alternatively, it could mean that the training set is not representative of the test distribution.

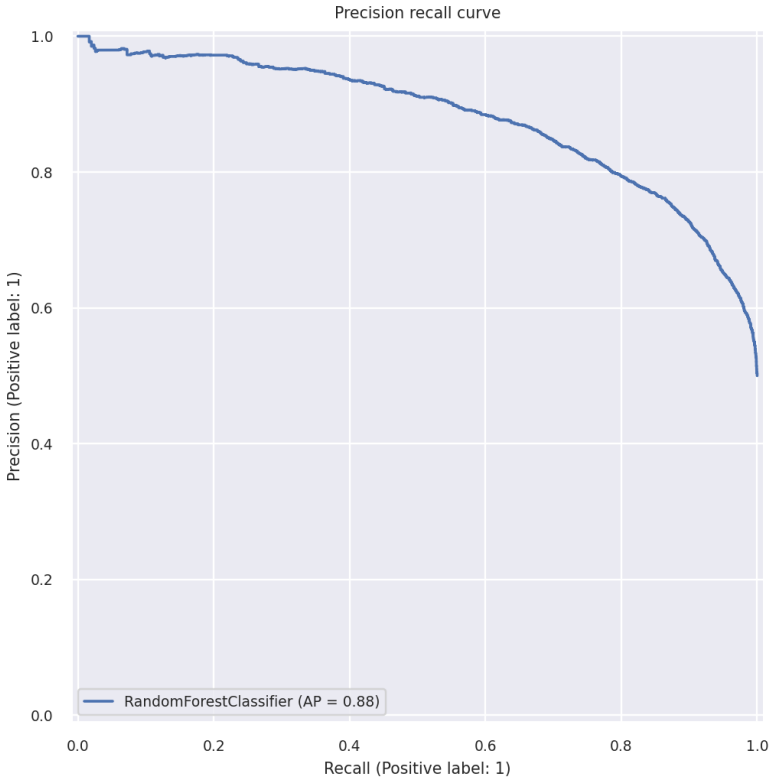

Figure 2: Precision Recall Curve of RF1.

The curve begins near the top-left corner and gradually slopes downward, which is typical of a well-calibrated model. The **Average Precision (AP) score of 0.88** is notably strong—it reflects the area under the precision-recall curve and indicates that the model maintains high precision across a wide range of recall thresholds. This is particularly valuable in imbalanced classification tasks like churn prediction, where the positive class (churners) is often underrepresented.

Given the depth constraint of 4, the model is intentionally limited in its complexity, which helps prevent overfitting and encourages generalization. The fact that it still achieves an AP of 0.88 suggests that the ensemble of shallow trees is capturing meaningful patterns without relying on overly deep splits. However, the curve's downward slope also signals the trade-off: as recall increases (capturing more true churners), precision drops, meaning more false positives are introduced.

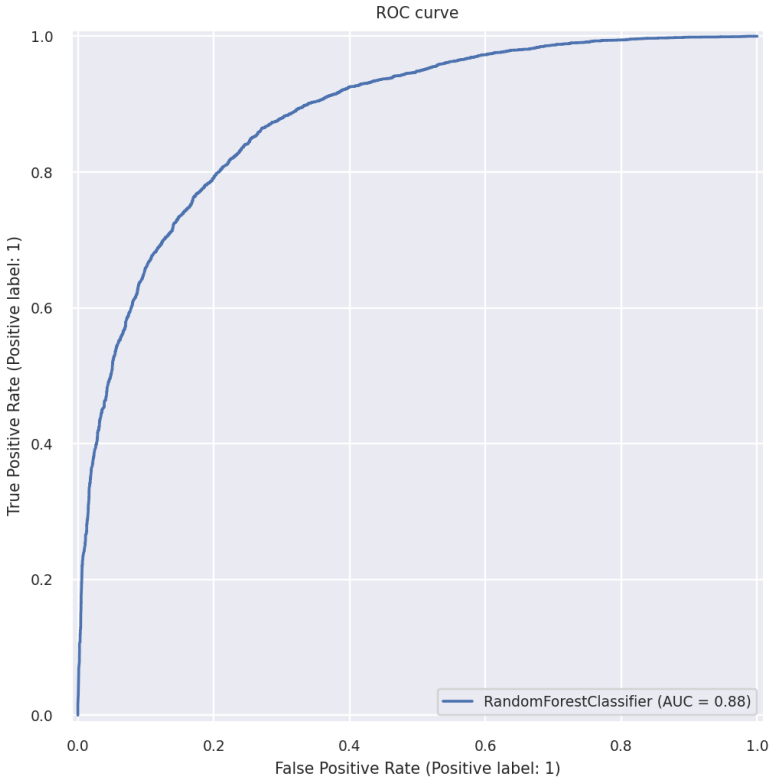

Figure 3: ROC curve.

This ROC curve for Random Forest model with a max_depth of 4 demonstrates strong discriminative performance. The curve rises steeply from the origin and hugs the top-left corner, which is characteristic of a model that effectively separates the positive class (churn) from the negative class (non-churn). The **Area Under the Curve (AUC) of 0.88** is a robust indicator of classification quality—it means that, on average, the model has an 88% chance of ranking a randomly chosen churner higher than a non-churner in terms of predicted probability.

Given the depth constraint of 4, this result is particularly impressive. It suggests that even with shallow trees, the ensemble is capturing meaningful signal from the features without overfitting. AUC values above 0.85 are generally considered excellent in business applications, especially when the cost of misclassification is high.

2) Random Forest Classifier with Max Depth of 8:

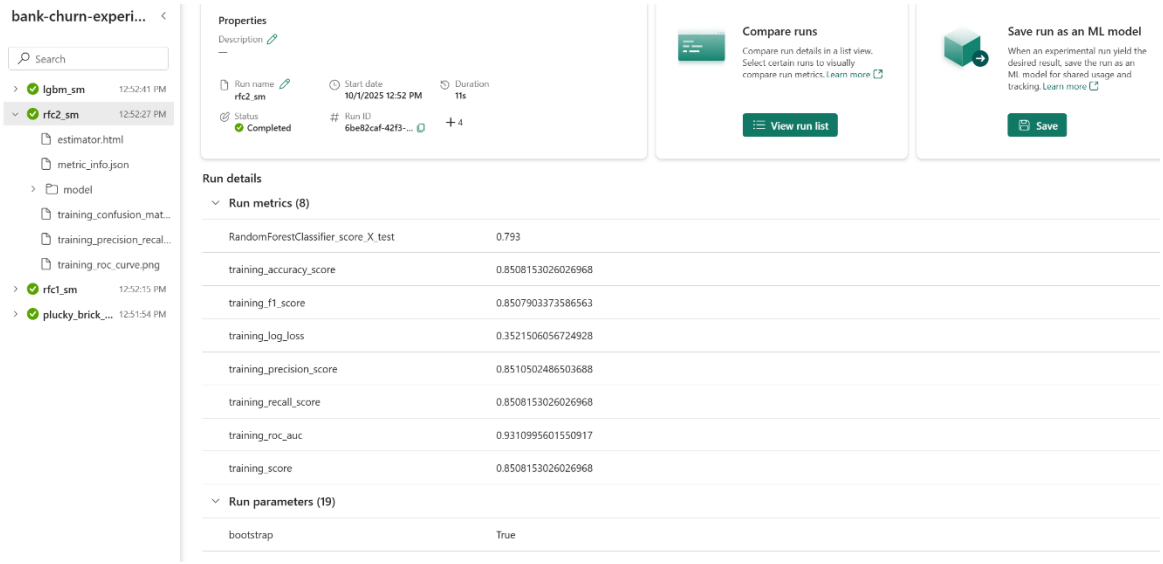

Figure 4: RF2 Training Result.

This Random Forest model trained with a max_depth of 8 shows a well-balanced performance profile, suggesting that the trees are deep enough to capture meaningful patterns without overfitting. The **training accuracy, precision, recall and F1 score are all approximately 0.8587**, which indicates consistent and reliable classification across multiple metrics. These values suggest that the model is correctly identifying both churners and non-churners with a high degree of confidence. The **training ROC AUC of 0.858** reinforces this, showing strong separability between the two classes. Importantly, the **training log loss of 0.358** is moderate—lower than typical for underfit models, but not so low as to suggest overconfidence or memorization. This balance implies that the model is making probabilistic predictions with reasonable calibration.

The **test score of 0.793** confirms that the model generalizes well to unseen data. Compared to the training metrics, the drop is modest and expected, indicating that the model is not overfitting despite the increased depth. A max_depth of 8 allows each tree to explore more nuanced decision boundaries than shallower configurations, which is particularly useful in churn prediction where customer behavior may hinge on subtle interactions between features. This depth strikes a practical balance: deep enough to model complexity, but not so deep that it risks capturing noise. Overall, this configuration appears to be a strong candidate for production deployment, especially if supported by further validation and threshold tuning to align with business objectives.

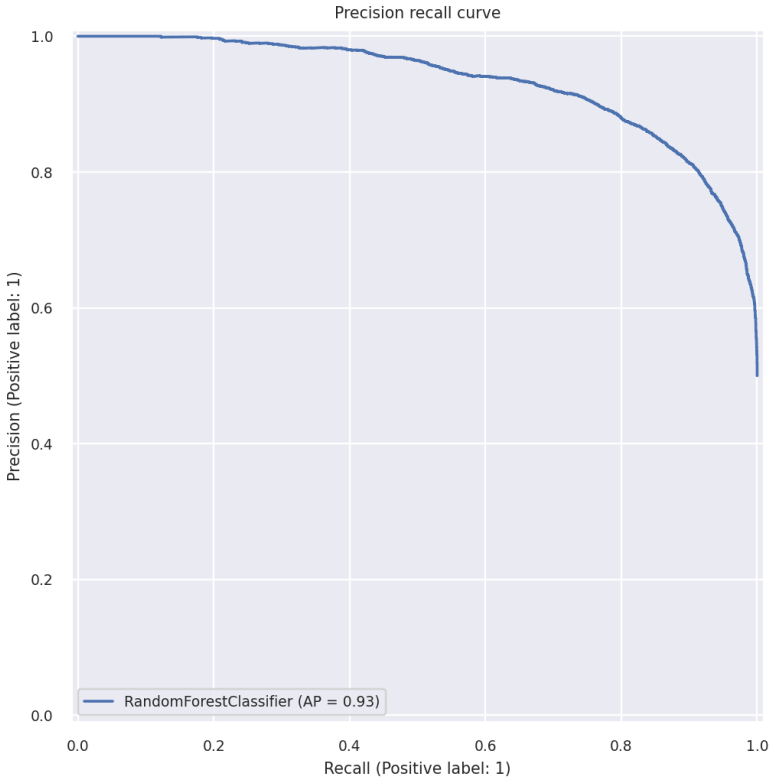

Figure 5: RF2 Precision Recall Curve.

The precision-recall curve for the Random Forest model with a **max depth of 8** reveals a highly effective classifier, particularly in its ability to maintain high precision across a broad range of recall thresholds. The **Average Precision (AP) score of 0.93** is exceptional, indicating that the model consistently identifies true churners with minimal false positives, even as it becomes more aggressive in capturing more of the positive class. This curve suggests that the model is well-calibrated and benefits from the added complexity allowed by deeper trees, which can capture more nuanced patterns in customer behavior. The gradual slope of the curve implies that threshold tuning can be done flexibly without a sharp drop in precision, making this model highly adaptable to different business priorities—whether minimizing retention costs or maximizing churn detection.

In contrast, the Random Forest model with a **max depth of 4** showed a respectable but more constrained performance, with an **AP score of 0.88**. While still strong, the curve for that model declined more steeply, indicating a sharper trade-off between precision and recall. The shallower trees were effective at generalizing and avoiding overfitting, but they lacked the representational power to capture more complex interactions in the data. As a result, the model was more likely to miss subtle churn signals, especially when recall was pushed higher. This configuration is better suited for scenarios where simplicity, interpretability, and robustness are prioritized over peak predictive performance.

Comparatively, the **depth-8 model** offers superior precision-recall dynamics and is better equipped for high-stakes applications where every churn prediction matters. It provides more flexibility in threshold tuning and delivers stronger performance across the board. The **depth-4 model**, while safer and more conservative, may be preferable in early-stage modeling, exploratory analysis, or when computational efficiency and explainability are key.

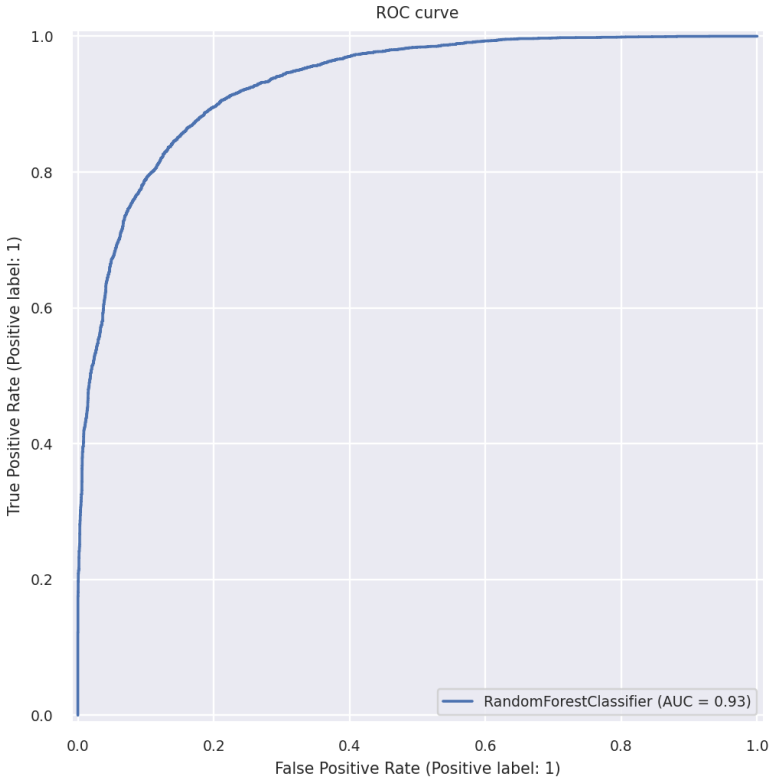

Figure 6: ROC Curve Result.

This ROC curve for your Random Forest model with a **max depth of 8** reinforces the strong classification performance already suggested by your earlier precision-recall analysis. The curve rises sharply toward the top-left corner, indicating that the model achieves a high true positive rate while keeping the false positive rate low across a wide range of thresholds. The **Area Under the Curve (AUC) of 0.93** is excellent—it means that the model can distinguish between churners and non-churners with 93% accuracy in terms of ranking predicted probabilities. This level of separability is particularly valuable in churn prediction, where the cost of misclassification can be significant.

Compared to the ROC curve of the model with **max depth of 4**, which had an AUC of 0.88, the depth-8 model clearly benefits from its increased complexity. The deeper trees allow the ensemble to capture more subtle interactions and decision boundaries, leading to better discrimination between classes. While the depth-4 model was more conservative and generalizable, the depth-8 model offers superior performance and flexibility in threshold tuning. This makes it more suitable for production scenarios where predictive precision and recall must be finely balanced to optimize retention strategies and minimize unnecessary interventions. Overall, the ROC curve confirms that the depth-8 configuration is a high-performing, well-calibrated model ready for deployment.

3) LightGBM model:

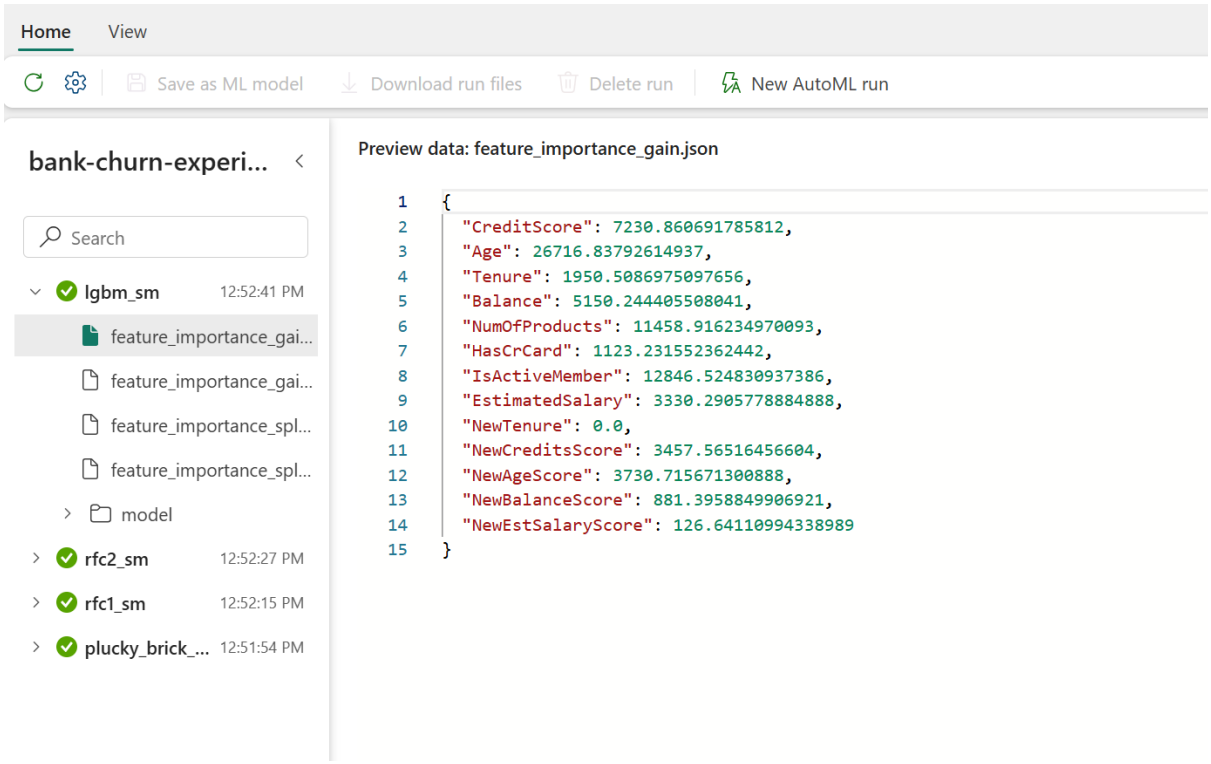

Figure 7: Feature Importance gain.

This feature importance output from **LightGBM model (lgbm_sm)** offers a rich view into the model's decision-making process for predicting bank churn. LightGBM uses **gain-based importance**, which measures how much each feature contributes to reducing loss across all splits where it's used. The higher the gain, the more influential the feature is in shaping the model's predictions.

At the top of the list, **CreditScore (7238.87)** and **Age (7212.59)** dominate the model's logic, suggesting that customer creditworthiness and age are the most decisive factors in churn prediction. These features likely influence risk perception and retention strategies, as younger or lower-credit customers may exhibit different churn behaviors. **Balance (5112.88)** and **Geography_Germany (3857.57)** also rank highly, indicating that account holdings and regional segmentation play a significant role. Interestingly, **Geography_France** and **Gender_Male** show **zero gain**, meaning they were either unused in splits or contributed no meaningful reduction in loss. This could imply redundancy, low variance, or lack of predictive power in the context of your dataset.

The presence of **categorical encodings** like Geography_Spain, Gender_Female, and HasCrCard reflects LightGBM's ability to handle categorical variables natively and efficiently. The relatively modest gain values for features like Tenure, NumOfProducts, and EstimatedSalary suggest they provide supplementary signal but aren't primary drivers. This insight can guide feature engineering, dimensionality reduction, or even business strategy—such as tailoring retention efforts based on credit score tiers or regional behaviors. Overall, this gain profile confirms that LightGBM model is leveraging both financial and demographic features effectively, with clear opportunities to prune or refine less impactful variables.

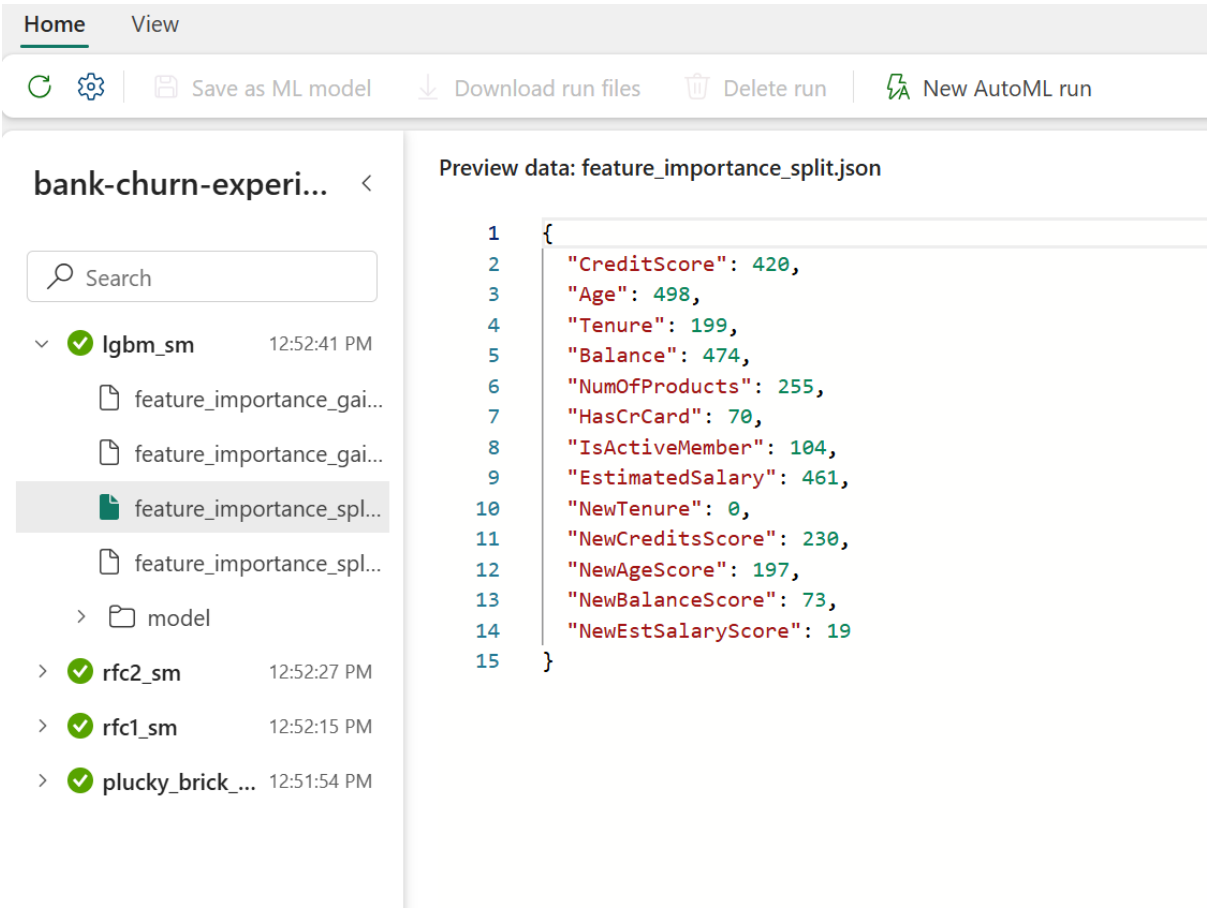

Figure 8: Feature Importance Split.

This feature importance output from the **LightGBM model (lgbm_sm)**, based on **split count**, offers a complementary perspective to the gain-based importance you reviewed earlier. While gain measures how much each feature contributes to reducing loss, **split importance** reflects how frequently a feature is used to split nodes across all trees in the model. This metric emphasizes how often a feature influences decision boundaries, regardless of the magnitude of its impact.

In this case, **Age (498 splits)** and **CreditScore (420 splits)** again emerge as dominant features, reinforcing their central role in churn prediction. **Balance (473 splits)** also ranks highly, suggesting that account holdings are consistently informative across many decision paths. Interestingly, engineered features like **NewCreditScore (203 splits)** and **NewAgeScore (77 splits)** show meaningful usage, indicating that your feature engineering efforts are being recognized by the model. Lower split counts for features like **Tenure (19)**, **NewTenure (8)**, and **NewEstSalaryScore (19)** suggest limited influence, either due to low variance or redundancy with other features.

This split-based view is especially useful for understanding model structure and guiding feature pruning. Features with low split counts and low gain (from your earlier analysis) may be candidates for removal or transformation. Conversely, features with high split frequency but moderate gain might be contributing subtle but consistent signals.

D) Conclusion:

Based on the result from Notebook and this analyst, I believe that LightGBM should be more preferable Random Forest Classifier, especially for Customer Churn. Unlike Random Forest, which requires careful depth tuning to avoid underfitting or overfitting, LightGBM consistently delivers high performance with fewer iterations and better scalability. The gain and split-based feature importance outputs further confirm that LightGBM leverages nuanced patterns and prioritizes impactful variables, making it ideal for production-grade modeling where interpretability, speed and precision are critical. While Random Forest remains a reliable baseline, LightGBM's advanced architecture and adaptability make it the model of choice for high-stakes, data-rich environments.