Customer Churn Prediction Report

Summary

This report provides an initial assessment of the customer churn prediction task based on the available data. The analysis indicates that the current dataset exhibits weak predictive features for distinguishing between churn and non-churn instances. As of now, no single feature stands out as a strong separator of churn and non-churn customers.

Data Evaluation

The dataset used for customer churn prediction was analyzed to assess the potential of its features for predicting customer churn. The evaluation involved exploratory data analysis and visualization of key features.

Data Details

The dataset includes the following columns:

- **CustomerID** and **Name**: These are identifiers for customers and are not contributing to predictive power.
- **Age**, **Gender**, **Location**, **Subscription_Length_Months**, **Monthly_Bill**, and **Total_Usage_GB**: These features are currently showing extremely weak predictive signals.

Data Preprocessing

In the preprocessing phase, the following steps were applied:

- **Handling Missing Values:** Any missing values in the dataset were addressed through appropriate techniques, ensuring data integrity.
- **Encoding Categorical Variables:** Categorical variables such as **Gender** and **Location** were encoded to numerical representations for machine learning compatibility.
- **Feature Scaling:** The features were scaled to ensure that they all contribute equally to the predictive model.

Feature Engineering

Feature engineering involved the following actions:

- **Age Binning:** The **Age** feature was binned into categories to capture different customer age groups and potentially enhance predictive power.
- **Feature Ratios:** Ratios were computed between features such as **Monthly_Bill** and **Total_Usage_GB** to capture insights related to usage habits.

For more comprehensive details on data preprocessing, feature engineering, and model development, refer to the accompanying **Python notebook**.

Model Performance: Logistic Regression

- **Accuracy:** The model's accuracy is approximately 50.4%, indicating that around 50.4% of the predictions made by the model are correct.

- **Precision:** The precision is approximately 50.0%. When the model predicts a customer will churn, it's correct about 50.0% of the time.
- **Recall:** The recall is approximately 38.98%. The model correctly identifies about 38.98% of the actual churn cases.
- **F1 Score:** The F1 score is approximately 0.438, indicating a balance between precision and recall.
- **ROC AUC Score:** The ROC AUC score is approximately 0.503, suggesting that the model's ability to distinguish between classes is only slightly better than random chance.

Overall as expected these metrics indicate that the model's performance is not very strong. The accuracy is close to random guessing.

Confusion Matrix Interpretation

The confusion matrix for the Logistic Regression model is as follows:

	Predicted Not Churn	**Predicted Churn**
Actual Not Churn	6213 (TN)	3866 (FP)
Actual Churn	6054 (FN)	3867 (TP)

In this matrix:

- True Negatives (TN): 6213 - The number of instances correctly predicted as "non-churn."
- False Positives (FP): 3866 - The number of instances incorrectly predicted as "churn" when they actually didn't.
- False Negatives (FN): 6054 - The number of instances incorrectly predicted as "non-churn" when they actually churned.
- True Positives (TP): 3867 - The number of instances correctly predicted as "churn."

Model Performance: XGBoost

- **Accuracy:** An accuracy of 0.50 indicates that the model's predictions are on par with random guessing. This means the model is not effectively distinguishing between churn and non-churn customers.

- **AUC-ROC Score:** With an AUC-ROC score of 0.50, the model's ability to differentiate between churn and non-churn customers is similar to random chance. This suggests that the model's predicted probabilities for churn and non-churn are not good or discriminative.

Classification Report:

- **Precision:** The precision values of 0.50 for both classes mean that when the model predicts a customer will churn (positive class) or not churn (negative class), it is equally likely to be correct for both cases. This could indicate that the model's decision boundary is not effectively separating the two classes.

- **Recall:** The recall values of 0.60 for non-churn (class 0) and 0.40 for churn (class 1) imply that the model is better at identifying customers who will not churn. However, it's less effective at identifying customers who will actually churn. This could suggest that the model is biased toward the majority class (non-churn).

- **F1-score:** The F1-scores of 0.55 for non-churn and 0.44 for churn further highlight the model's struggle to balance precision and recall for both classes.
- **Support:** The support values (number of instances) for each class indicate that the dataset has a roughly equal number of both churn and non-churn instances. This suggests that class imbalance is not the primary issue affecting the model's performance.

Confusion Matrix Interpretation

The confusion matrix for the XGBoost model is as follows:

	Predicted Not Churn	**Predicted Churn**
Actual Not Churn	6000 (TN)	4079 (FP)
Actual Churn	5916 (FN)	4005 (TP)

In this matrix:

- **True Negatives (TN):** 6000 - The number of instances correctly predicted as "non-churn." These are customers who were correctly identified as not likely to churn.
- **False Positives (FP):** 4079 - The number of instances incorrectly predicted as "churn" when they actually didn't. These are cases where the model falsely identified customers as likely to churn when they didn't.
- **False Negatives (FN):** 5916 - The number of instances incorrectly predicted as "non-churn" when they actually churned. These are instances where the model failed to identify customers who actually ended up churning.
- **True Positives (TP):** 4005 - These are instances that were correctly predicted as "churn." In the context of churn prediction, these are customers who were correctly identified as likely to churn.

To summarize:

- **True Negatives (TN):** 6000 - Correctly predicted "non-churn" customers.
- **False Positives (FP):** 4079 - Incorrectly predicted "churn" when it's actually "non-churn."
- **False Negatives (FN):** 5916 - Incorrectly predicted "non-churn" when it's actually "churn."
- **True Positives (TP):** 4005 - Correctly predicted "churn" customers.

Overall as expected these metrics indicate that the xgboost model's performance is not very strong. The accuracy is close to random guessing.

Logistic Regression vs. XGBoost: A Comparison of Confusion Matrices

Logistic Regression Confusion Matrix:

	Predicted Not Churn	Predicted Churn
Actual Not Churn	6213 (TN)	3866 (FP)
Actual Churn	6054 (FN)	3867 (TP)

XGBoost Confusion Matrix:

	Predicted Not Churn	Predicted Churn
Actual Not Churn	6000 (TN)	4079 (FP)
Actual Churn	5916 (FN)	4005 (TP)

In comparing these two confusion matrices:

- For Logistic Regression, we observe 6213 true negatives (TN), 3866 false positives (FP), 6054 false negatives (FN), and 3867 true positives (TP).
- For XGBoost, we have 6000 true negatives (TN), 4079 false positives (FP), 5916 false negatives (FN), and 4005 true positives (TP).

Conclusion: Logistic Regression vs. XGBoost

Comparing the performance of Logistic Regression and XGBoost models based on their confusion matrices, we can draw the following observations:

- **True Negatives (TN):** Both models have relatively similar TN values, with Logistic Regression having a slightly higher count of correct "non-churn" predictions.
- **False Positives (FP):** XGBoost has a higher count of FP, indicating that it has falsely predicted more instances as "churn" when they were actually "non-churn."
- **False Negatives (FN):** Logistic Regression has a higher count of FN, meaning it has incorrectly predicted more instances as "non-churn" when they were actually "churn."
- **True Positives (TP):** XGBoost has a slightly higher count of TP, correctly identifying more instances as "churn."

Both models exhibit similar strengths and weaknesses in their ability to predict churn. While Logistic Regression shows higher TN and FN values, indicating a tendency to predict more instances as "non-churn," XGBoost has a higher rate of false positives, indicating a tendency to over-predict "churn."

In the context of customer churn prediction, both models struggle to find the right balance between precision and recall.

Conclusion

The initial analysis of the provided dataset suggests that the predictive power of the current features for customer churn prediction is limited. The absence of strong distinguishing factors presents challenges for building an effective predictive model. Even after extensive feature engineering efforts, the models' performances have not shown substantial improvement. challenges for building an effective predictive model.