Telco Customer Churn - Classification Report

Sakka Mohamad-Mario 1241EB – Subgroup II

Dataset: https://www.kaggle.com/datasets/blastchar/telco-customer-churn

1. OBJECTIVE

I built a supervised learning pipeline to predict whether a customer will churn. I cleaned the dataset, created a preprocessing pipeline for numeric and categorical variables, trained several classifiers, and evaluated them using churn-focused metrics and confusion matrices. I then repeated the same process using SMOTE to rebalance the training data and compared results.

2. DATASET AND CLASS IMBALANCE

I used the Kaggle Telco Customer Churn dataset. The target variable is Churn (Yes/No). In my held-out test split, there are 1033 non-churn customers and 372 churn customers (churn rate ≈ 26.5%). This class imbalance matters because accuracy can look strong even when the model misses many churners.

3. CLEANING AND PREPARATION

I cleaned the dataset by removing customerID, converting TotalCharges to numeric values, removing duplicates, and ensuring missing values are handled consistently. I kept imputation, scaling, and one-hot encoding inside pipelines to avoid leakage.

4. FEATURE RELATIONSHIP CHECK

I examined correlation among numeric features to identify redundancy. The strongest relationship appears between tenure and TotalCharges, which is expected because TotalCharges accumulates as tenure increases. This supports dropping TotalCharges if I want a simpler feature set.

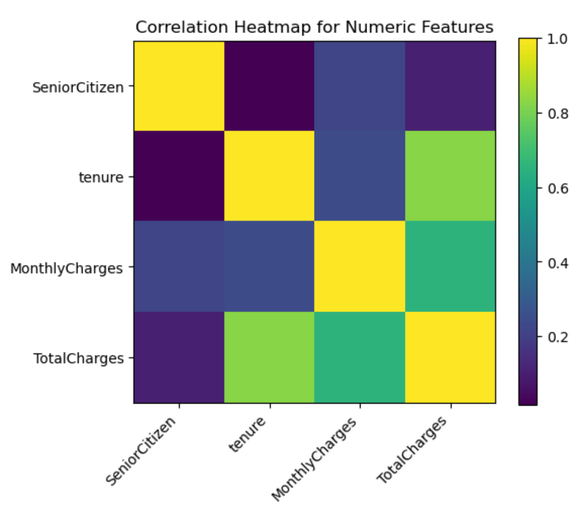

Figure 1. Correlation heatmap for numeric features.

5. MODELS AND PREPROCESSING

I used a ColumnTransformer to preprocess features: numeric features are imputed with the median and scaled, and categorical features are imputed with the most frequent value and one-hot encoded. I trained Logistic Regression, SVM (RBF), Decision Tree, Random Forest, KNN, and Gaussian Naive Bayes.

6. BASELINE RESULTS

I evaluated each model on the held-out test set. Because churn is the minority class, I focus on churn recall and churn F1 in addition to accuracy. Confusion matrices help me inspect churn false negatives directly.

Table 1. Baseline test-set metrics.

Model	**Acc**	**Bal. Acc**	**Prec**	**Rec**	**F1**
Gaussian Naive Bayes	0.686	0.734	0.450	0.836	0.585
KNN	0.785	0.707	0.605	0.540	0.571
Logistic Regression	0.803	0.714	0.661	0.524	0.585
Decision Tree	0.794	0.701	0.639	0.505	0.565
SVM (RBF)	0.799	0.699	0.663	0.487	0.561
Random Forest	0.779	0.673	0.612	0.449	0.518

In the baseline run, Logistic Regression achieved the highest accuracy, while Gaussian Naive Bayes achieved the highest churn recall but with lower precision. Several models still miss a meaningful portion of churners, which motivates rebalancing.

6.1 BASELINE CONFUSION MATRICES

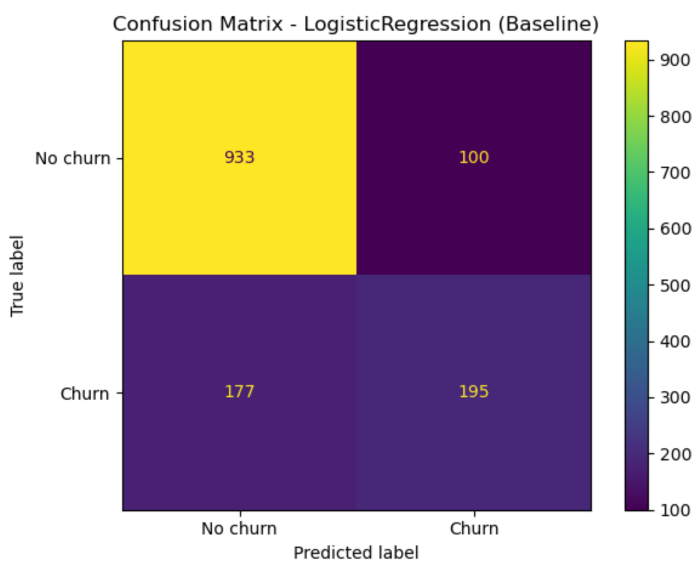

Figure 2. Confusion matrix for Logistic Regression (baseline).

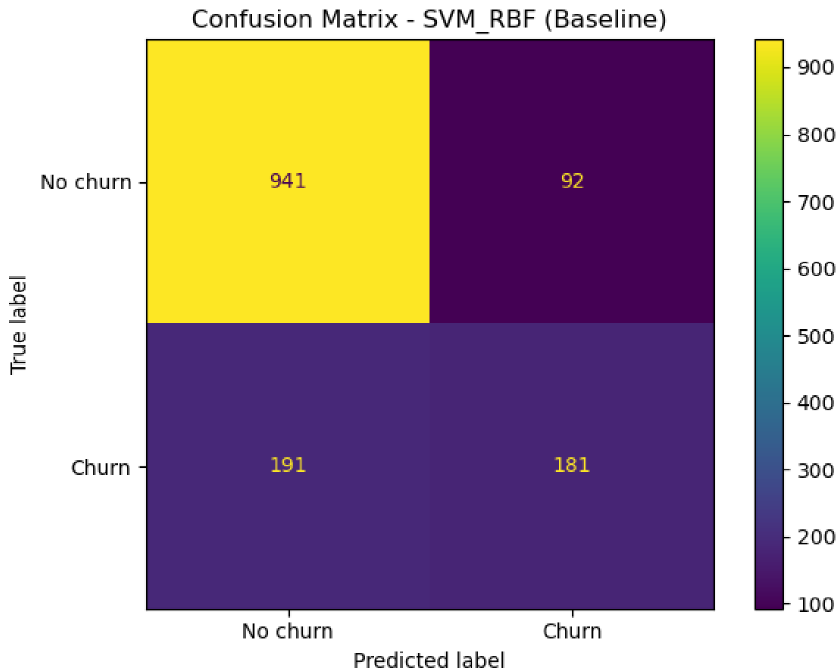

Figure 3. Confusion matrix for SVM (RBF) (baseline).

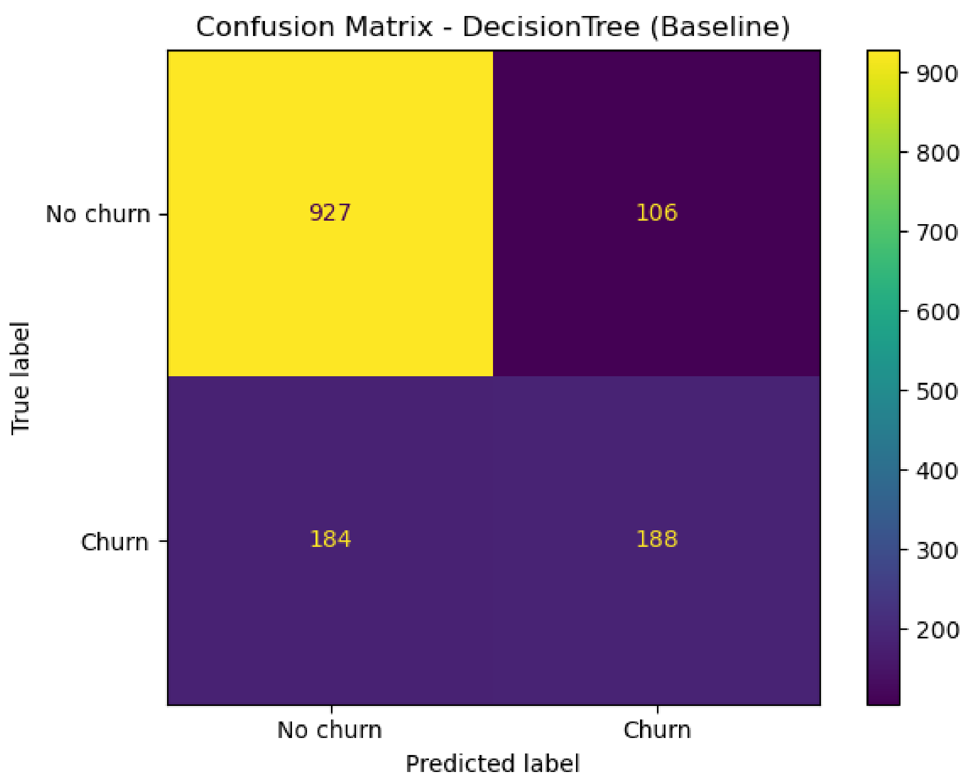

Figure 4. Confusion matrix for Decision Tree (baseline).

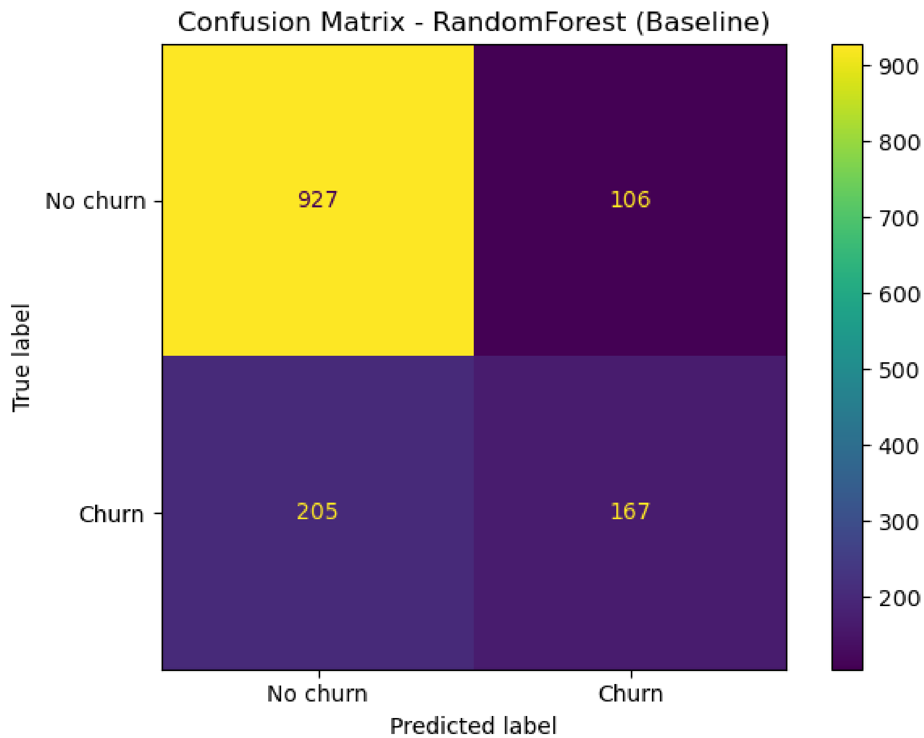

Figure 5. Confusion matrix for Random Forest (baseline).

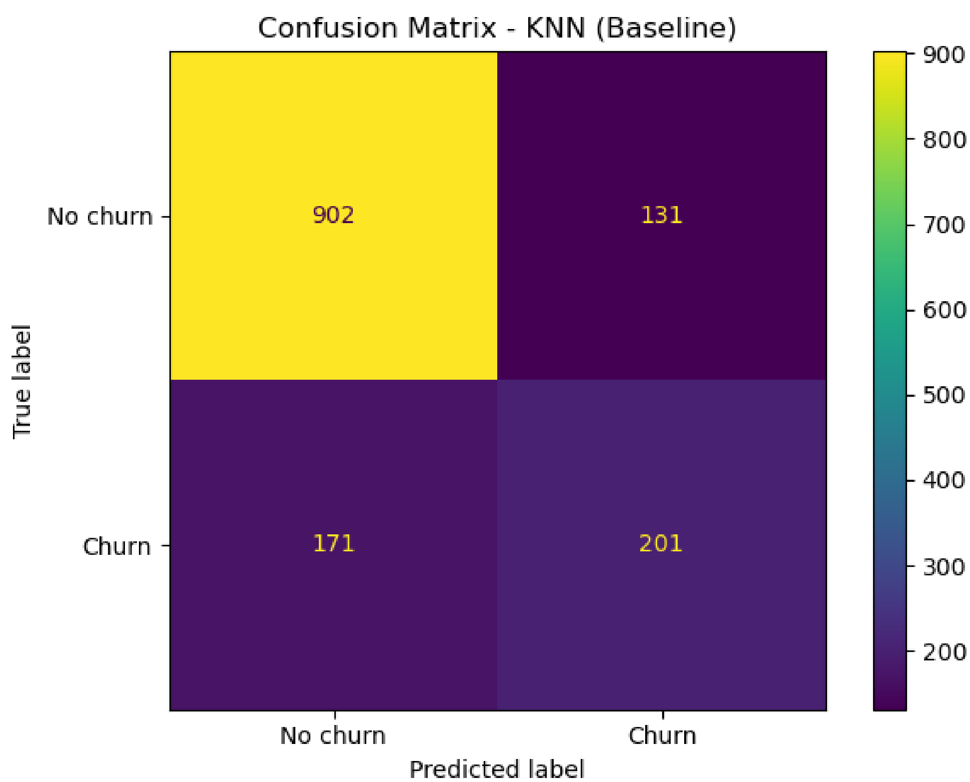

Figure 6. Confusion matrix for KNN (baseline).

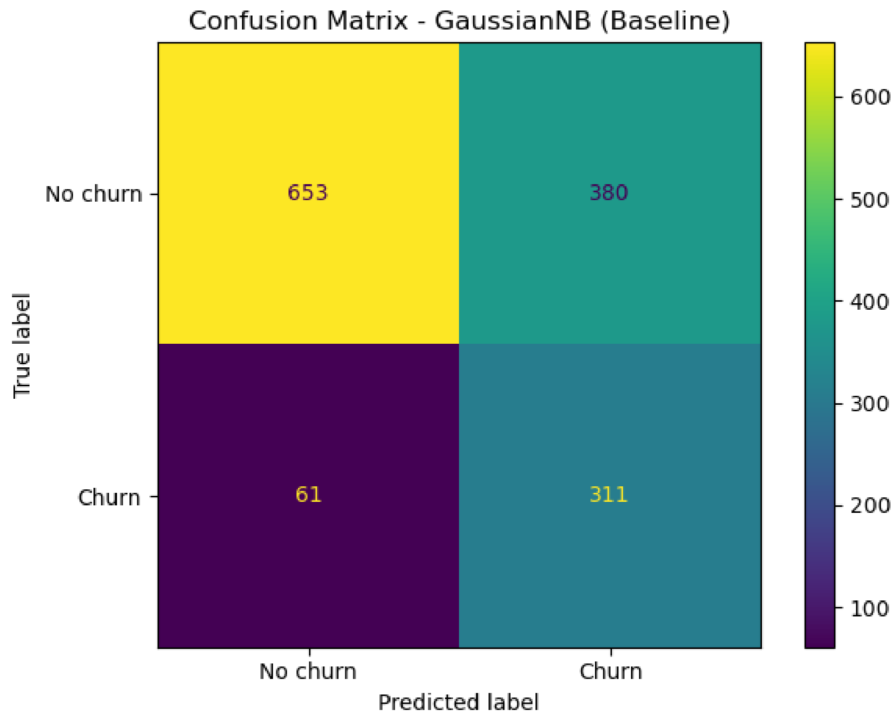

Figure 7. Confusion matrix for Gaussian Naive Bayes (baseline).

6.2 BASELINE CHURN-METRIC BAR CHARTS

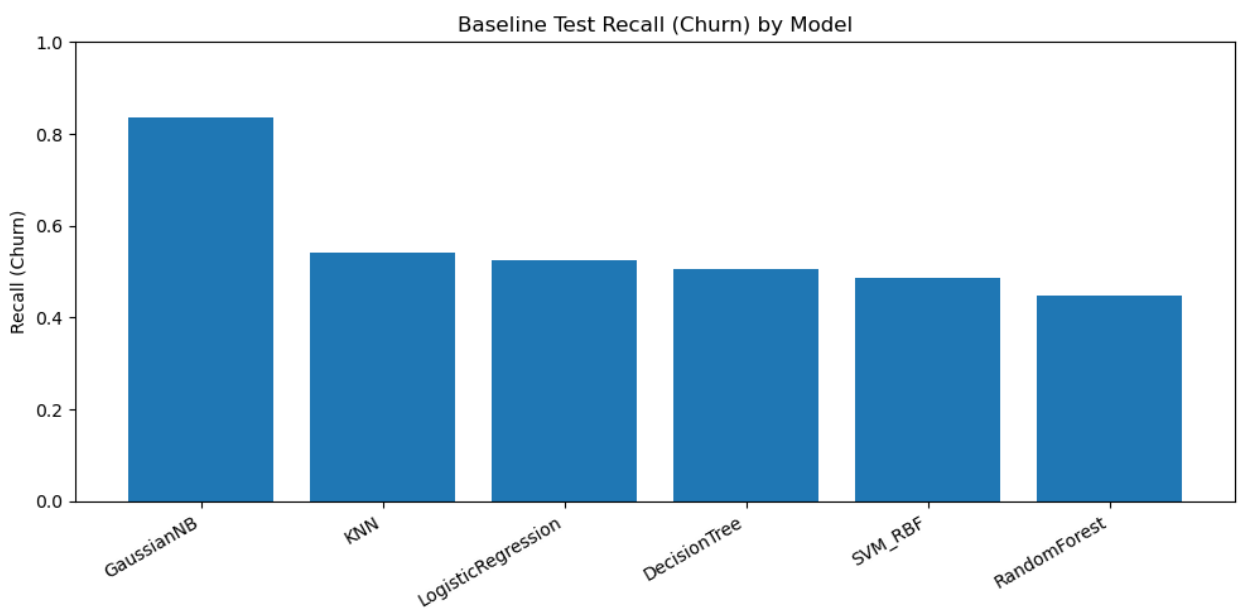

Figure 8. Baseline test recall for churn by model.

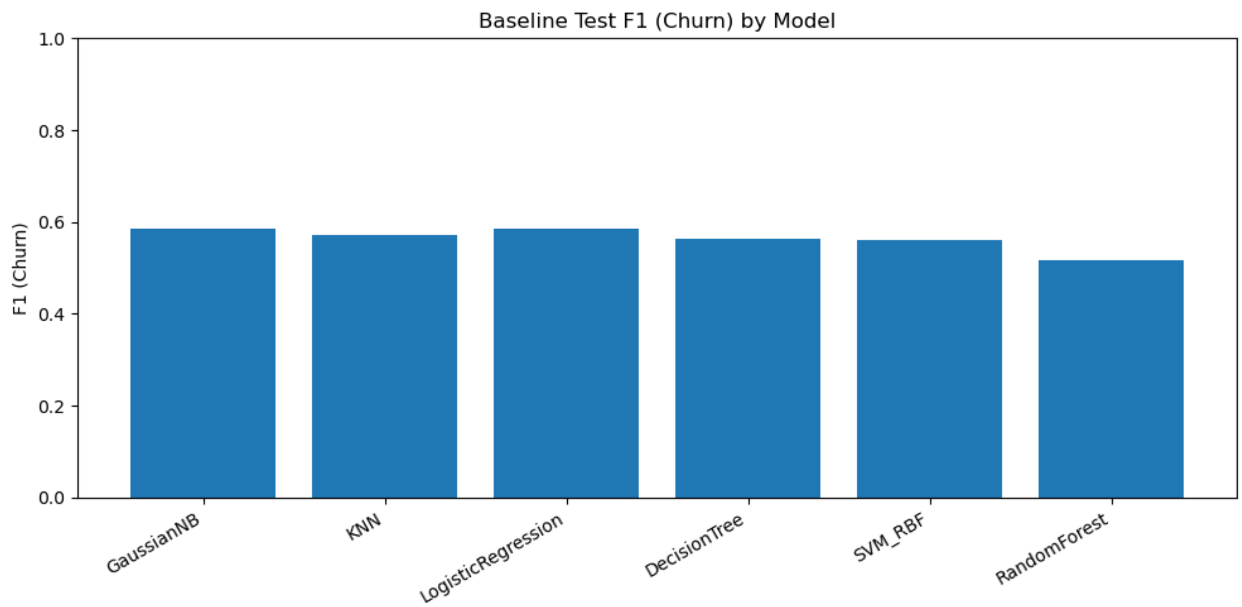

Figure 9. Baseline test F1 for churn by model.

7. BASELINE CROSS-VALIDATION

I used stratified 5-fold cross-validation to estimate how stable each model is across different splits. I report mean performance across folds.

Table 2. Baseline 5-fold cross-validation means.

Model	CV Acc	CV Bal. Acc	CV Rec	CV F1
Gaussian Naive Bayes	0.693	0.742	0.847	0.594
KNN	0.787	0.716	0.564	0.584
Logistic Regression	0.803	0.720	0.545	0.594
Decision Tree	0.788	0.701	0.515	0.563
SVM (RBF)	0.799	0.698	0.482	0.559
Random Forest	0.788	0.690	0.481	0.545

7.1 BASELINE CV CHARTS

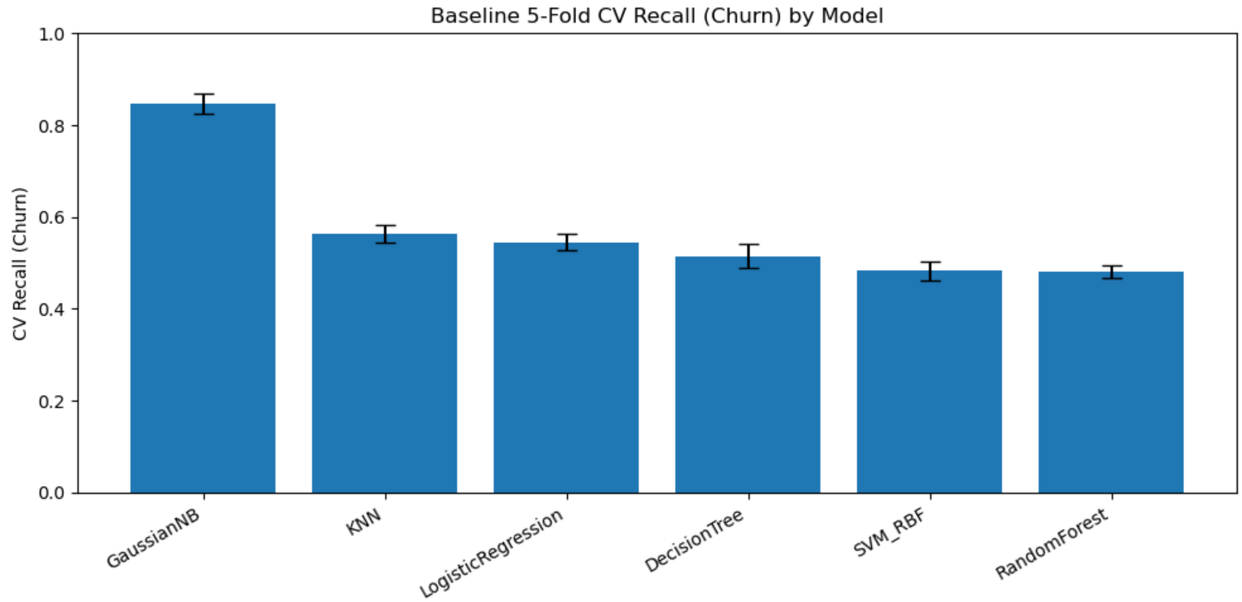

Figure 10. Baseline 5-fold CV recall for churn by model.

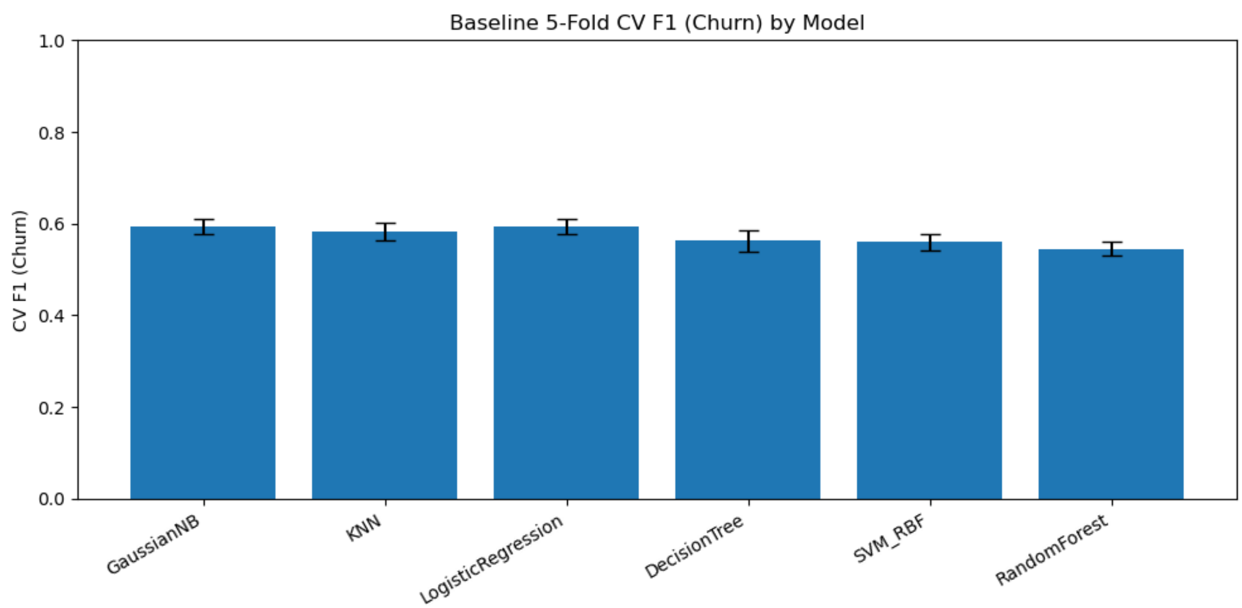

Figure 11. Baseline 5-fold CV F1 for churn by model.

8. SMOTE REBALANCING

To improve churn detection, I applied SMOTE to oversample churn cases in the training data while keeping the test set unchanged. I consider SMOTE beneficial when churn recall and churn F1 increase, even if accuracy decreases.

Table 3. SMOTE test-set metrics.

Model	Acc	Bal. Acc	Prec	Rec	F1
Gaussian Naive Bayes	0.700	0.742	0.463	0.831	0.594
KNN	0.706	0.740	0.468	0.812	0.594
Logistic Regression	0.749	0.753	0.517	0.763	0.617
Decision Tree	0.758	0.743	0.532	0.710	0.608
SVM (RBF)	0.774	0.749	0.560	0.694	0.619
Random Forest	0.773	0.694	0.578	0.527	0.551

With SMOTE, churn recall improves noticeably for Logistic Regression, SVM (RBF), Decision Tree, and KNN. The tradeoff is more false positives and lower accuracy for some models.

8.1 SMOTE CONFUSION MATRICES

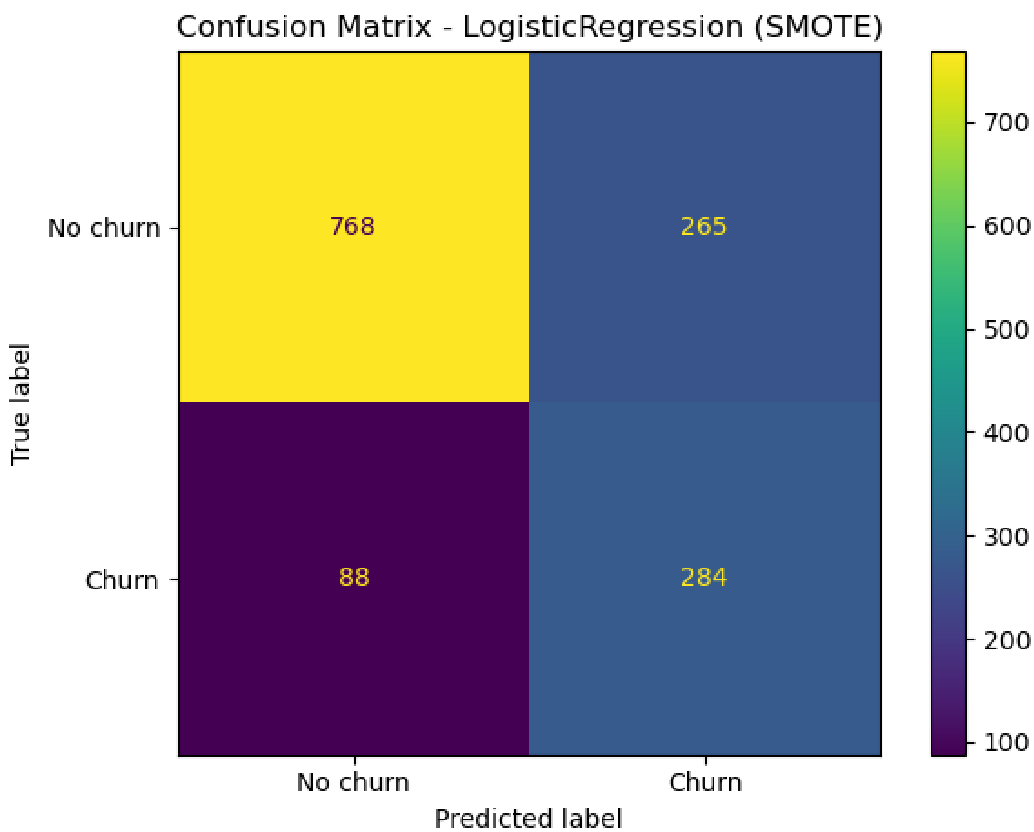

Figure 12. Confusion matrix for Logistic Regression (SMOTE).

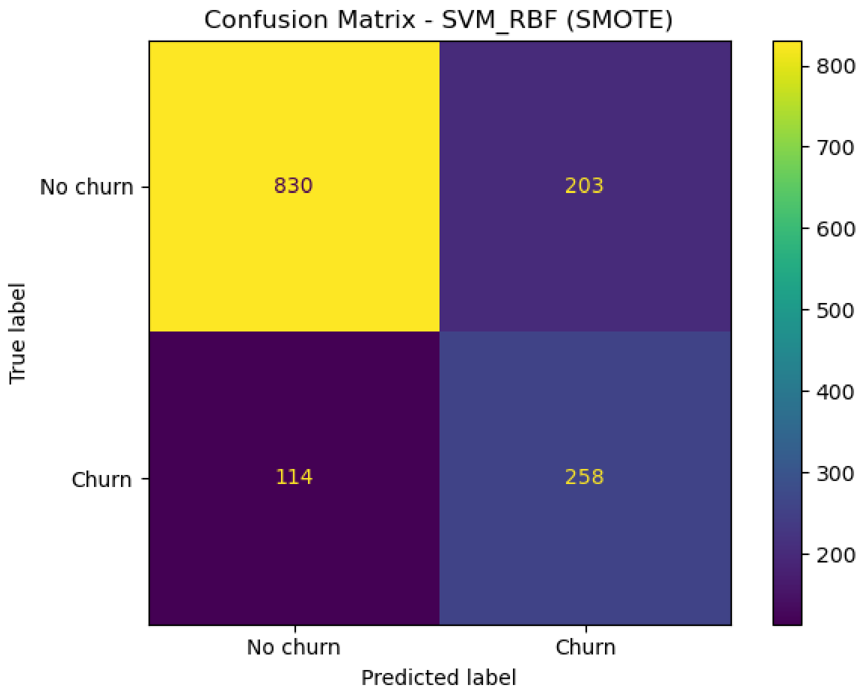

Figure 13. Confusion matrix for SVM (RBF) (SMOTE).

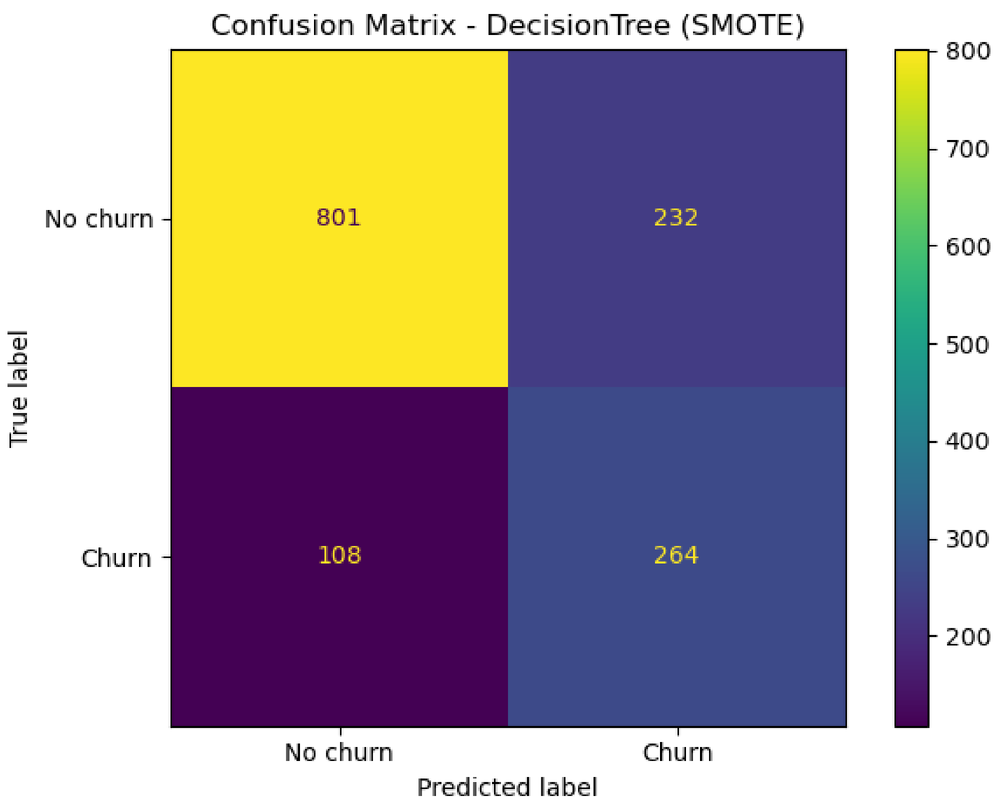

Figure 14. Confusion matrix for Decision Tree (SMOTE).

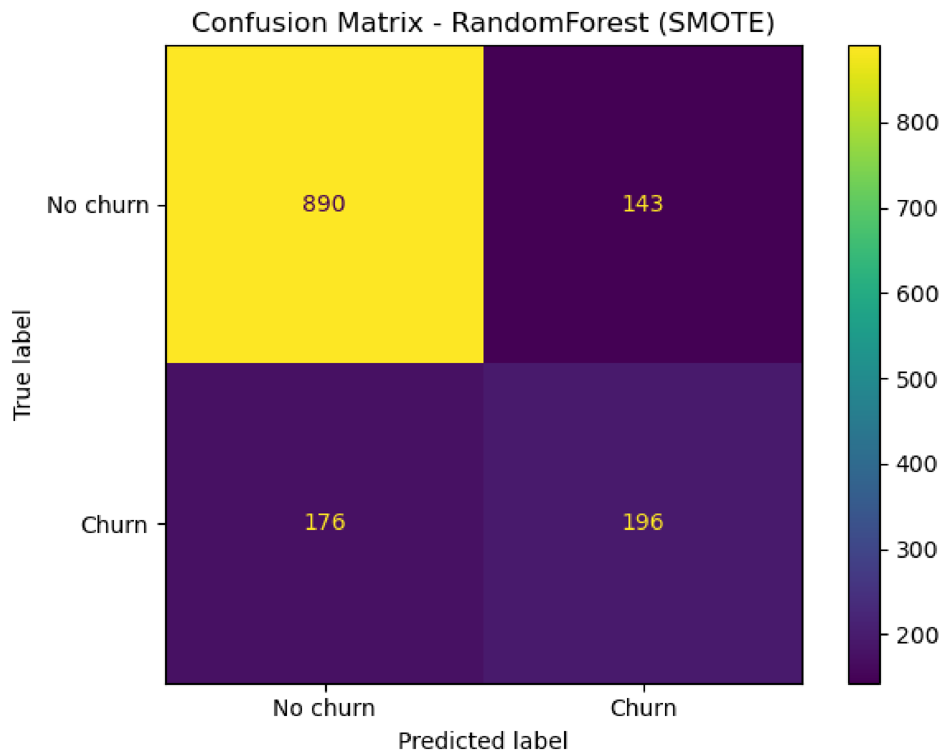

Figure 15. Confusion matrix for Random Forest (SMOTE).

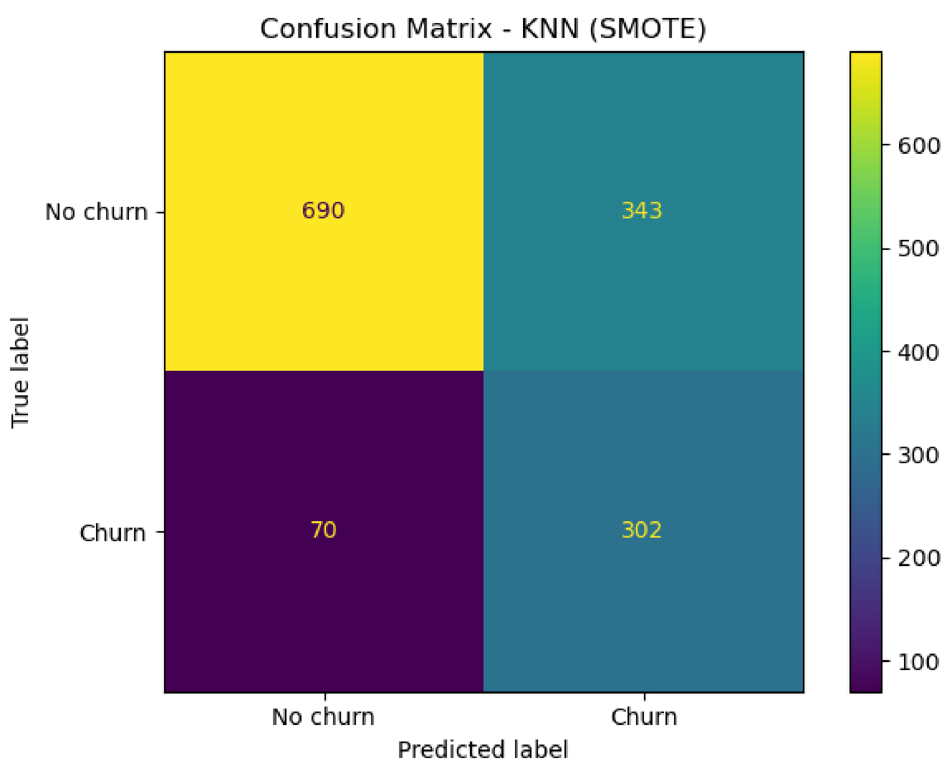

Figure 16. Confusion matrix for KNN (SMOTE).

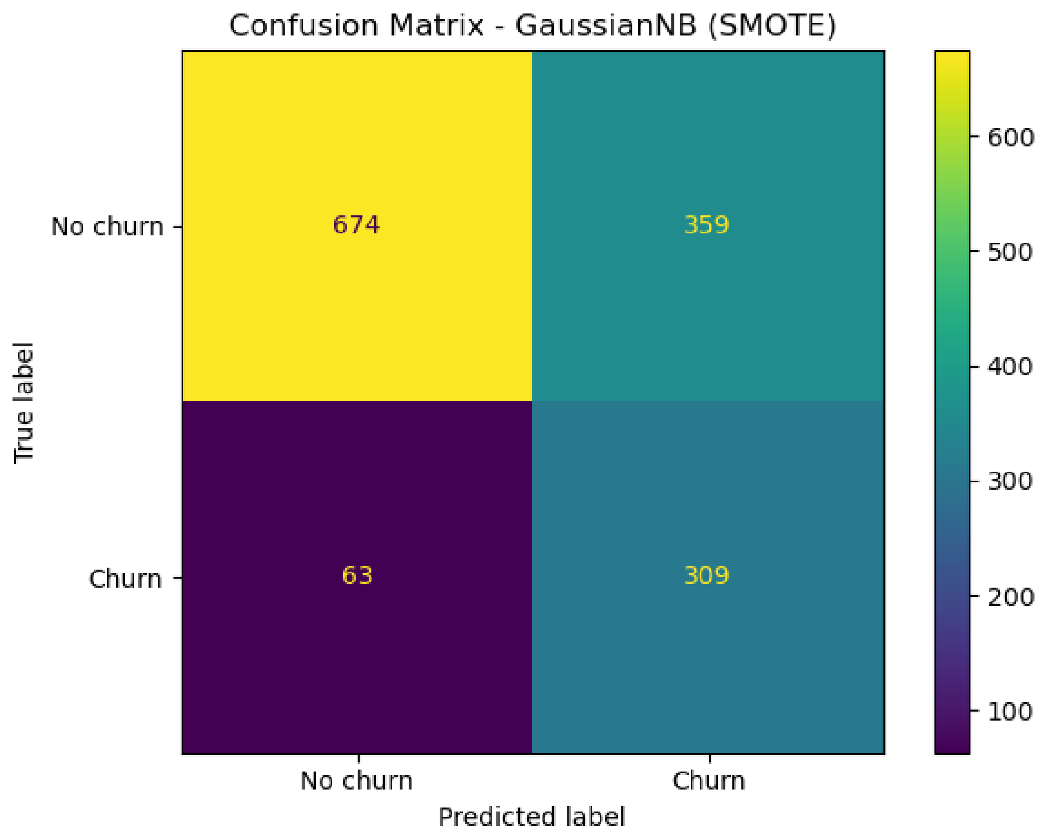

Figure 17. Confusion matrix for Gaussian Naive Bayes (SMOTE).

8.2 TEST-SET COMPARISON: BASELINE VS SMOTE

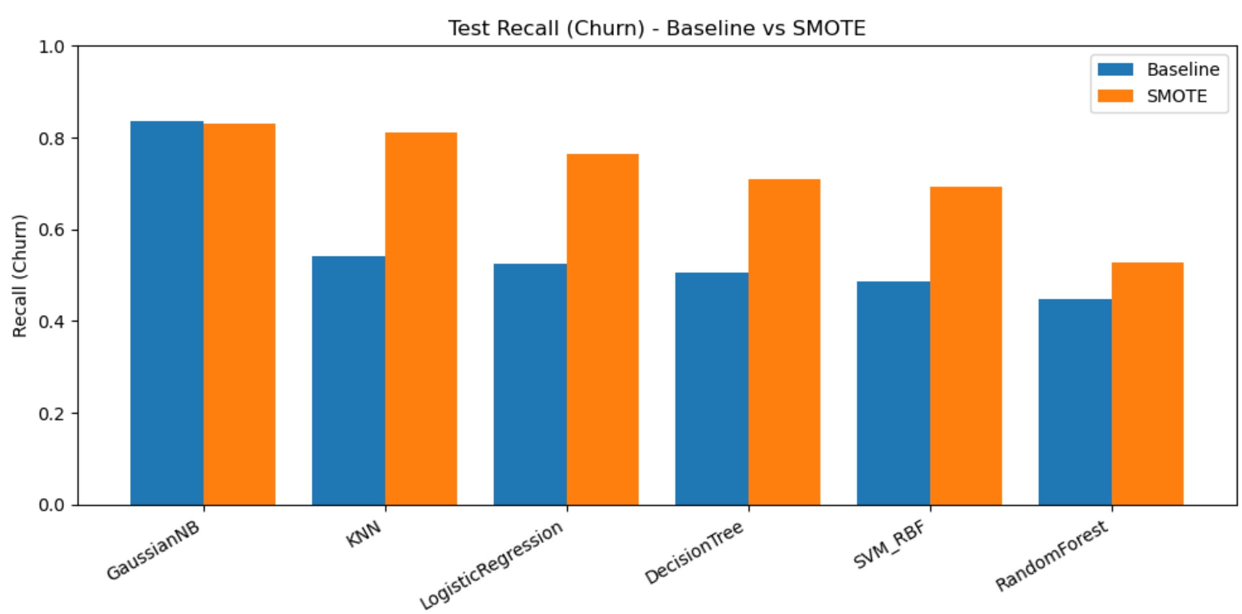

Figure 18. Test recall for churn: baseline vs SMOTE.

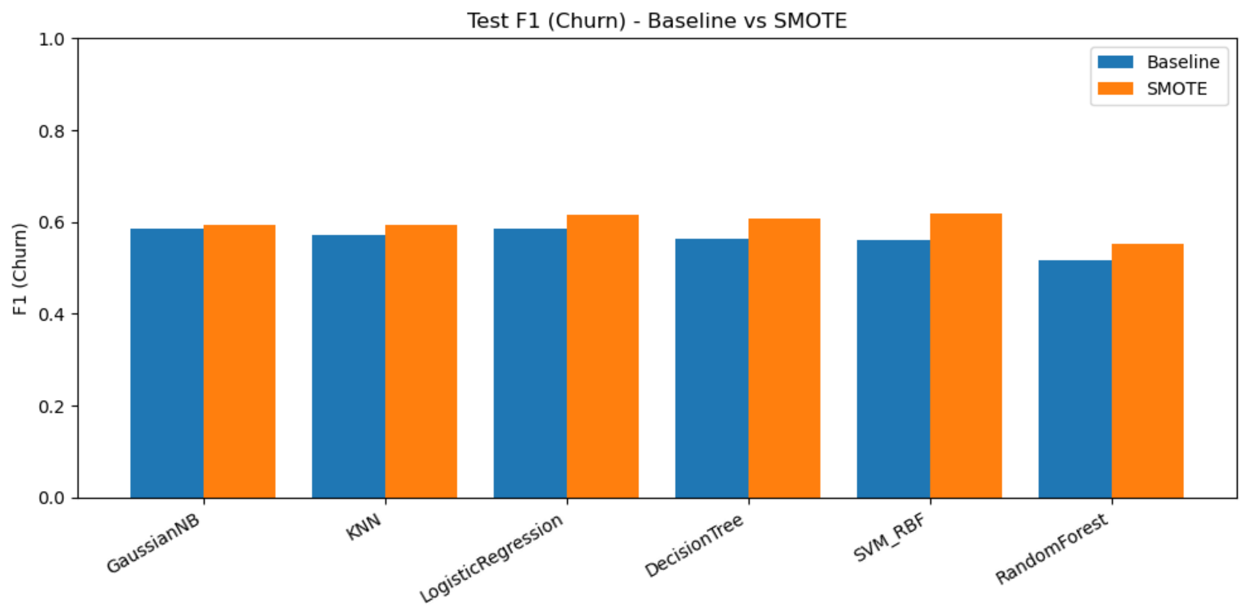

Figure 19. Test F1 for churn: baseline vs SMOTE.

Table 4. Test-set churn recall and F1 comparison (baseline vs SMOTE).

Model	**Rec Base**	**Rec SMOTE**	**F1 Base**	**F1 SMOTE**
Gaussian Naive Bayes	0.836	0.831	0.585	0.594
KNN	0.540	0.812	0.571	0.594
Logistic Regression	0.524	0.763	0.585	0.617
Decision Tree	0.505	0.710	0.565	0.608
SVM (RBF)	0.487	0.694	0.561	0.619
Random Forest	0.449	0.527	0.518	0.551

9. CROSS-VALIDATION COMPARISON

I also compared baseline vs SMOTE using stratified 5-fold cross-validation. For SMOTE, oversampling happens only within each training fold to avoid leakage.

Table 5. Cross-validation churn recall and F1 comparison (baseline vs SMOTE).

Model	**CV Rec Base**	**CV Rec SMOTE**	**CV F1 Base**	**CV F1 SMOTE**
Gaussian Naive Bayes	0.847	0.838	0.594	0.599
KNN	0.564	0.827	0.584	0.590
Logistic Regression	0.545	0.798	0.594	0.633
Decision Tree	0.515	0.706	0.563	0.611
SVM (RBF)	0.482	0.727	0.559	0.621
Random Forest	0.481	0.555	0.545	0.572

9.1 CROSS-VALIDATION COMPARISON CHARTS

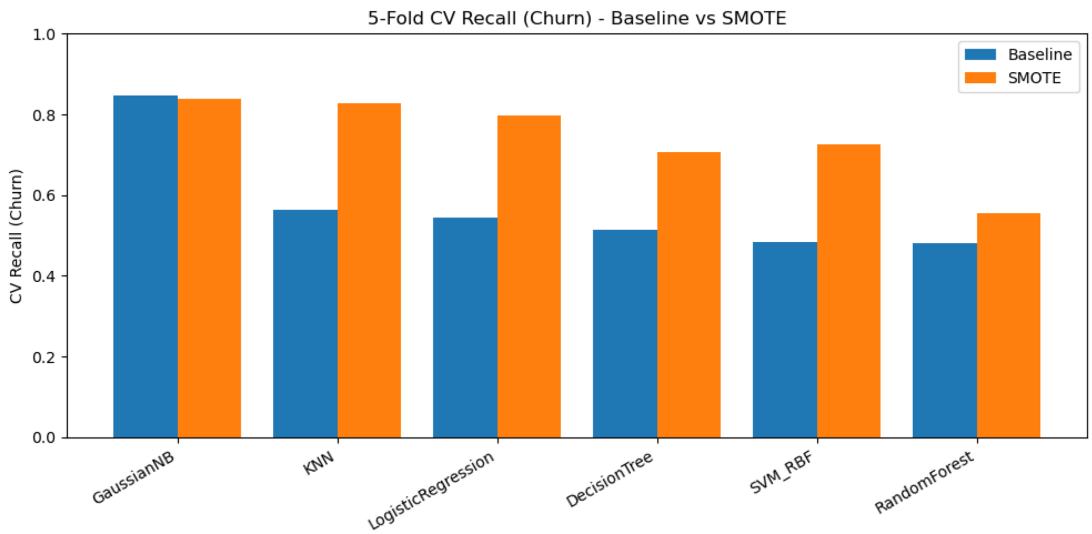

Figure 20. 5-fold CV recall for churn: baseline vs SMOTE.

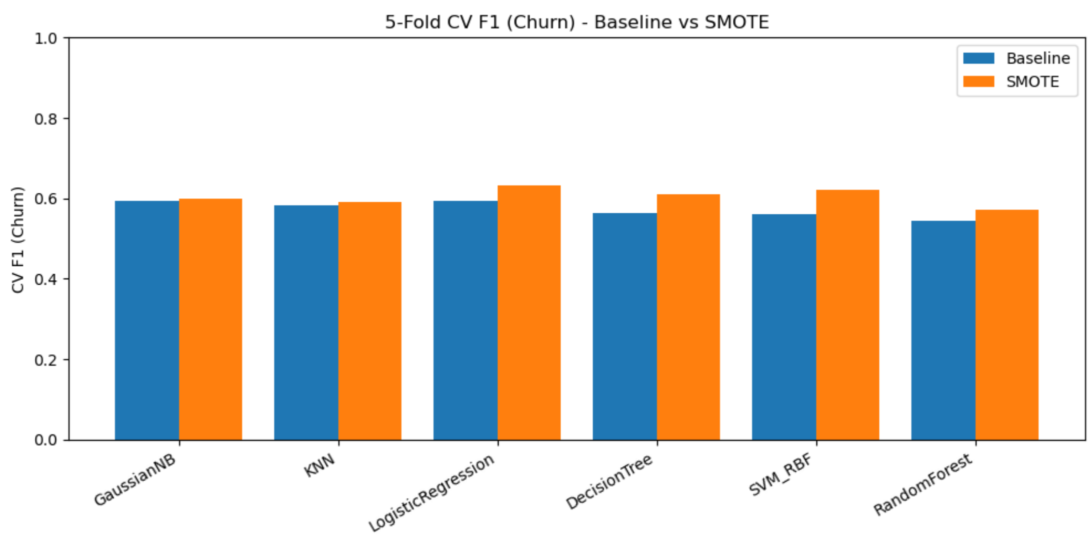

Figure 21. 5-fold CV F1 for churn: baseline vs SMOTE.

10. CONCLUSION

The baseline models reach good accuracy, but churn recall is the main limitation. SMOTE improves churn recall and churn F1 for most models, meaning I catch more churners at the cost of more false positives and sometimes lower accuracy. If the goal is retention targeting, I prefer the SMOTE results because they reduce churn false negatives. Next, I would try threshold tuning or class-weighted training to control the precision-recall tradeoff based on business costs.