Telco Customer Churn

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

1. OVERVIEW

I trained several standard classifiers to predict whether a customer will churn. I focused on cleaning the dataset, checking feature relationships, and evaluating model performance with confusion matrices on a held-out test set.

2. DATA CLEANING AND PREPARATION

I cleaned the dataset by removing the customer identifier, converting TotalCharges into a numeric field, removing duplicates, and ensuring the dataset was ready for modeling. After converting TotalCharges, blanks can become missing values, so I handled any missing values using simple rules that keep the preprocessing consistent.

3. CORRELATION ANALYSIS

The correlation heatmap below shows the linear relationships among the numeric features. The strongest relationship is between tenure and TotalCharges, which makes sense because TotalCharges accumulates over time. MonthlyCharges is also moderately related to TotalCharges. SeniorCitizen has weak correlation with the other numeric fields.

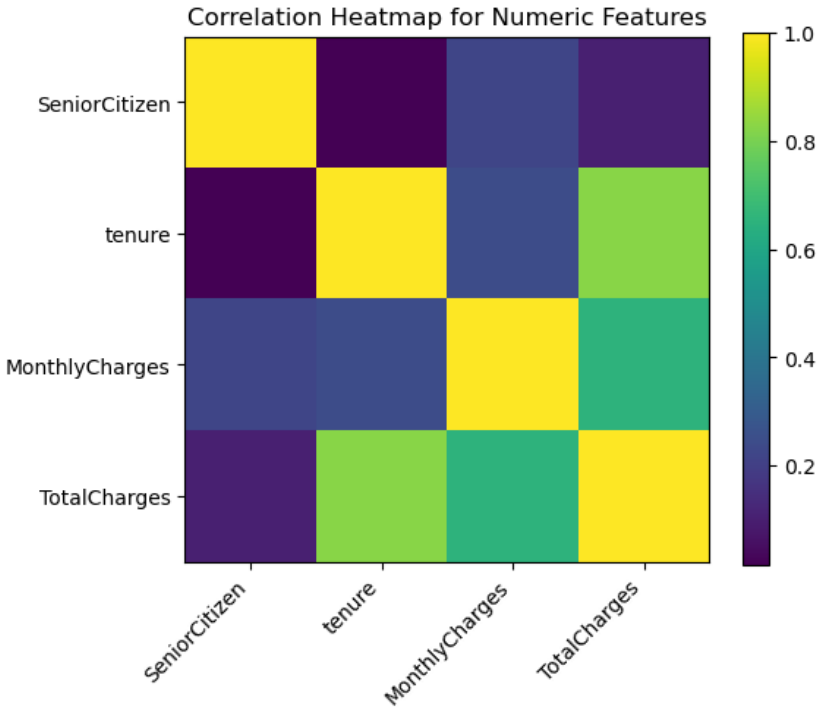

Figure 1. Correlation heatmap for numeric features.

Based on this, TotalCharges is the most likely candidate for removal if I want to reduce redundancy, since it overlaps with tenure and MonthlyCharges. I can keep it if the goal is purely predictive performance, but dropping it can simplify the feature space without losing much information.

4. TEST SET EVALUATION

I evaluated each model on the same test set. Each confusion matrix is organized as follows: top-left is true negatives, top-right is false positives, bottom-left is false negatives, and bottom-right is true positives. The churn class is the positive class.

5. CONFUSION MATRICES

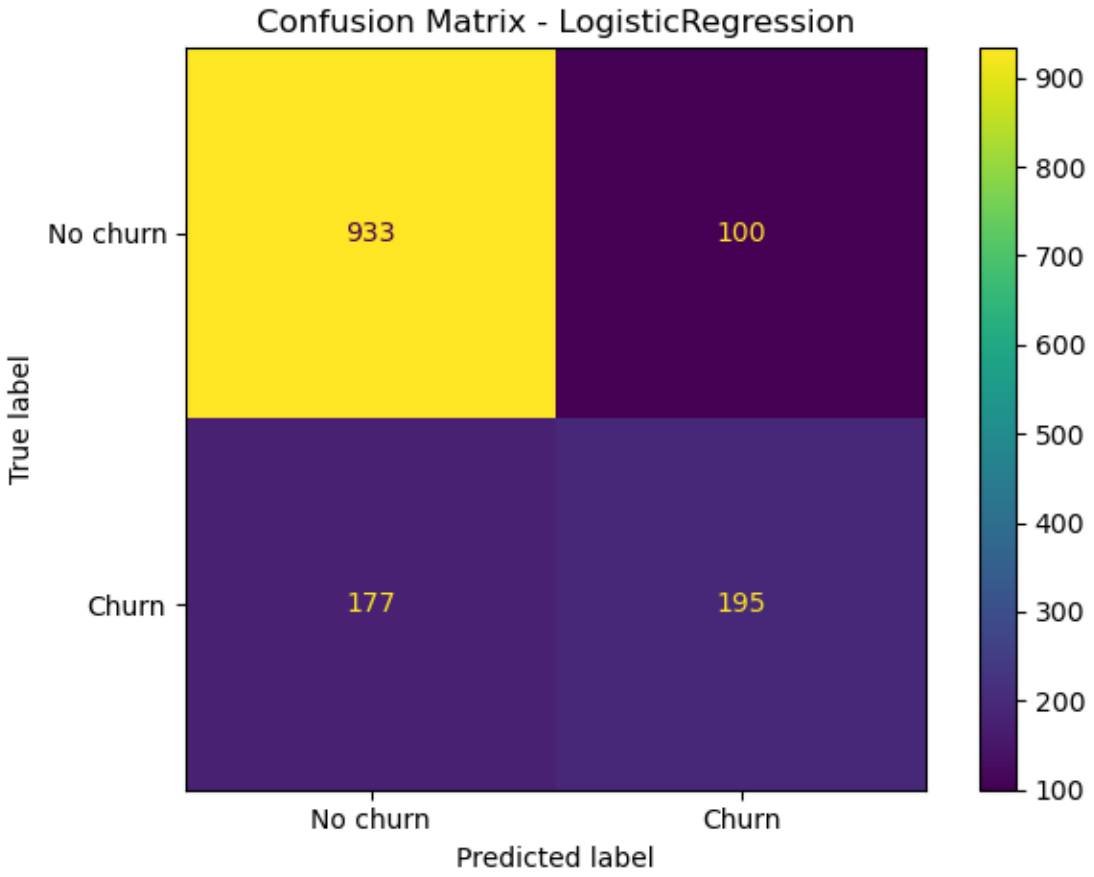

Figure 2. Confusion matrix for Logistic Regression

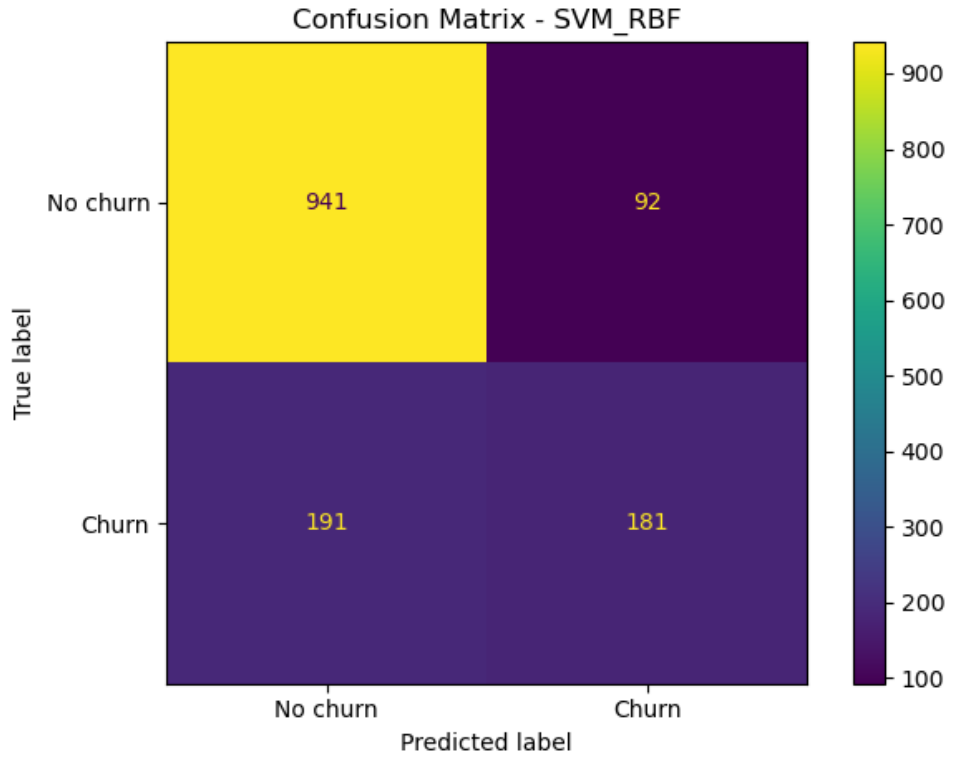

Figure 3. Confusion matrix for SVM (RBF)

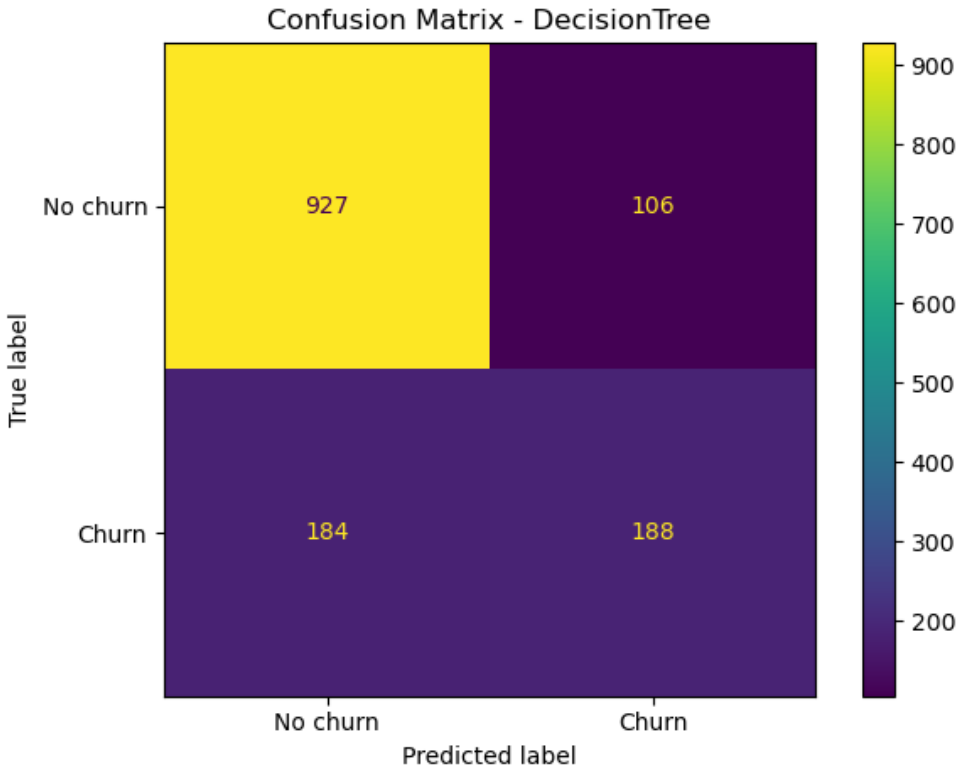

Figure 4. Confusion matrix for Decision Tree

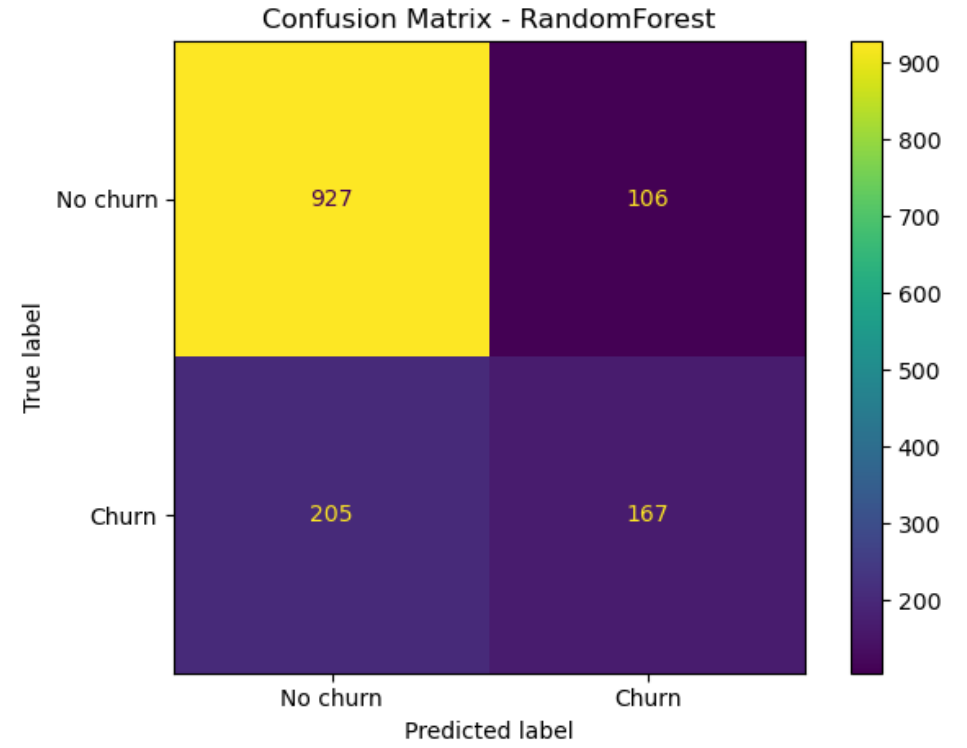

Figure 5. Confusion matrix for Random Forest

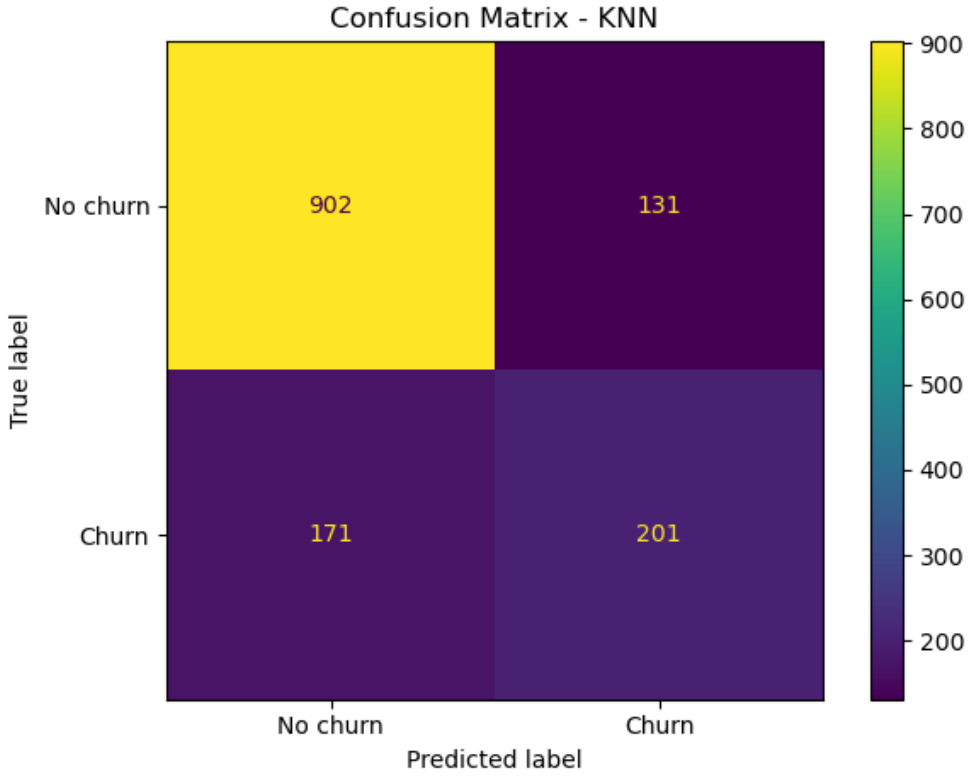

Figure 6. Confusion matrix for KNN

6. METRICS SUMMARY FROM CONFUSION MATRICES

The table below summarizes accuracy and churn-focused precision/recall/F1 computed directly from the confusion matrices shown in the notebook outputs.

Model	Accuracy	Precision (Churn)	Recall (Churn)	F1 (Churn)
Logistic Regression	0.803	0.661	0.524	0.585
SVM (RBF)	0.799	0.663	0.487	0.561
Decision Tree	0.794	0.639	0.505	0.565
KNN	0.785	0.605	0.540	0.571
Random Forest	0.779	0.612	0.449	0.518

Across these runs, the test-set accuracy is around 0.78 to 0.80 for most models. The confusion matrices also show that the dataset is imbalanced, with many more non-churn customers than churn customers in the test split. For example, in the Logistic Regression matrix there are 5174 no-churn cases compared to 1869 churn cases. Recall for churn is noticeably lower than precision, which indicates the models miss a meaningful portion of churners. This is typical when the dataset has fewer churners than non-churners and the decision threshold remains at the default.

7. INTERPRETATION OF RESULTS

Logistic Regression achieved the highest accuracy among the shown confusion matrices, and it also provides a good baseline because it is easy to interpret and tune. SVM performed similarly but did not improve churn recall. KNN produced a slightly better churn recall than most of the other models in this run, but at the cost of more false positives.

The consistent pattern across models is that false negatives are high relative to true positives. If the goal is to identify churners for retention campaigns, I would likely tune for higher recall. Practical ways to do this include adjusting the decision threshold using predicted probabilities, using class weights, or optimizing for F1 or recall instead of accuracy.

8. CONCLUSION

My numeric correlation check shows a strong relationship between tenure and TotalCharges, so dropping TotalCharges is reasonable if I want a simpler feature set. In terms of predictive performance, all tested models are in a similar range on accuracy, but they tend to miss churners. The next improvement I would make is to tune the model for higher churn recall and evaluate the tradeoff between catching more churners and increasing false positives.