

# Customer Churn Prediction Using Machine Learning

Monika Tyagi  
Indian Institute of Science  
Email: monikatyagi@iisc.ac.in

Sourajit Bhar  
Indian Institute of Science  
Email: sourajitbhar@iisc.ac.in

**Abstract**—Customer churn prediction is a critical task for subscription-based businesses. In this project, we analyze the Telco Customer Churn dataset and apply multiple supervised machine learning models — including Logistic Regression, Decision Tree, Random Forest, Naive Bayes, and Support Vector Machine (SVM) — to detect at-risk customers. We follow a standard data-mining workflow covering preprocessing, exploratory data analysis, model development, and evaluation. Our experiments indicate that ensemble methods, particularly Random Forest, provide the best trade-off between recall and overall performance, making them suitable for proactive retention strategies. Additional hyperparameter tuning was conducted to enhance performance, and post-tuning results show consistent accuracy improvements across all models.

**Index Terms**—Customer Churn, Classification, Data Mining, Machine Learning, Random Forest, Hyperparameter Tuning, Telco

## I. Introduction

Customer churn refers to the phenomenon in which customers discontinue a service. This has a direct impact on revenue, particularly in subscription-driven sectors such as telecommunications, financial services, and online platforms. Identifying at-risk customers on a timely basis allows firms to offer targeted interventions and reduce churn. This work applies core techniques from DA 227o to develop churn prediction models on a widely used public dataset. Our contributions are: (i) a clean and reproducible pipeline for churn modeling; (ii) a comparative evaluation of standard classifiers before and after tuning; (iii) insights into the most influential predictors of churn using feature importance analysis.

## II. Related Work

Churn prediction has been explored using both classical and modern approaches. Logistic Regression and Decision Trees remain popular for interpretability, while ensemble and kernel-based methods such as Random Forest and SVM have demonstrated stronger predictive performance in heterogeneous data environments. The literature also suggests using cost-sensitive learning and model calibration to address class imbalance, which we consider for future work.

## III. Dataset

We use the *Telco Customer Churn* dataset (Kaggle), which contains 7,043 customer records with demographic details such as gender, age range, and whether they have partners and dependents; service usage features (phone, internet, online security, online backup, device protection, tech support, streaming services); account information (tenure, contract,

payment method, paperless billing, monthly and total charges); and churn labels. The target variable is binary: Churn = Yes/No. Data preprocessing included:

- Removal of redundant columns (e.g., customerID)
- Handling of missing values in TotalCharges
- One-hot encoding of categorical features
- Normalization of continuous attributes such as tenure and MonthlyCharges

## IV. Methodology

Our modeling pipeline follows a structured workflow:

- 1) **Data Cleaning and Preprocessing:** Data consistency checks, imputation, and feature encoding.
- 2) **Exploratory Data Analysis (EDA):** Distribution plots, correlation matrices, and churn percentage visualization.
- 3) **Feature Engineering:** Derived binary flags for service bundles and total monthly cost.
- 4) **Model Training:** Logistic Regression, Decision Tree, Random Forest, Naive Bayes, and SVM (RBF kernel).
- 5) **Evaluation:** Metrics include accuracy, precision, recall, F1-score, ROC–AUC, and confusion matrices.
- 6) **Hyperparameter Tuning:** Conducted using GridSearchCV and cross-validation for optimal parameters.
- 7) **Evaluation of Tuned Models:** Evaluated the models on the same parameters after the Hyperparameter Tuning.
- 8) **Result Comparison:** Comparison of results obtained before and after Hyperparameter Tuning.

## V. Experiments and Results

### A. Initial Model Comparison

TABLE I: Model Performance (Before Tuning)

Model	Acc.	Prec.	Recall	F1	AUC
Logistic Regression	0.8031	0.6456	0.5749	0.6082	0.8363
Random Forest	0.7875	0.6222	0.5107	0.5609	0.8171
SVM (RBF)	0.7868	0.6259	0.4920	0.5509	0.7909
Decision Tree	0.7186	0.4701	0.4626	0.4663	0.6366
Naive Bayes	0.6439	0.4179	0.8636	0.5632	0.8105

- **Logistic Regression:** Best overall model before tuning. Highest accuracy and ROC-AUC; balanced precision and recall. A strong, interpretable baseline for churn prediction.
- **Random Forest:** A strong model with high potential but slightly underperforming before tuning. It handles non-linear relationships well and is robust to noise, but recall is

lower than Logistic Regression. Hyperparameter tuning (e.g., `max_depth`, `n_estimators`) can significantly boost its performance.

- **Support Vector Machine (SVM):** A solid model that performs well with scaled, high-dimensional data. It has good precision but relatively low recall. With kernel and regularization tuning, SVM can substantially improve its ROC-AUC and recall.
- **Decision Tree:** Prone to overfitting and weak generalization pre-tuning. It captures non-linear patterns but exhibits the lowest ROC-AUC, indicating poor discrimination. Tuning (e.g., `max_depth`, `min_samples_split`) is essential for improving generalization.
- **Naive Bayes:** Shows extremely high recall but very low precision meaning it identifies most churners but mislabels many non-churners. This makes it useful for early churn alerts but less reliable overall.

**Summary:** Linear models outperform non-linear ones before tuning. Recall remains the weakest metric; ROC-AUC is more reliable than accuracy. Naive Bayes identifies churners well but triggers many false positives.

#### B. Confusion Matrices (Before Tuning)

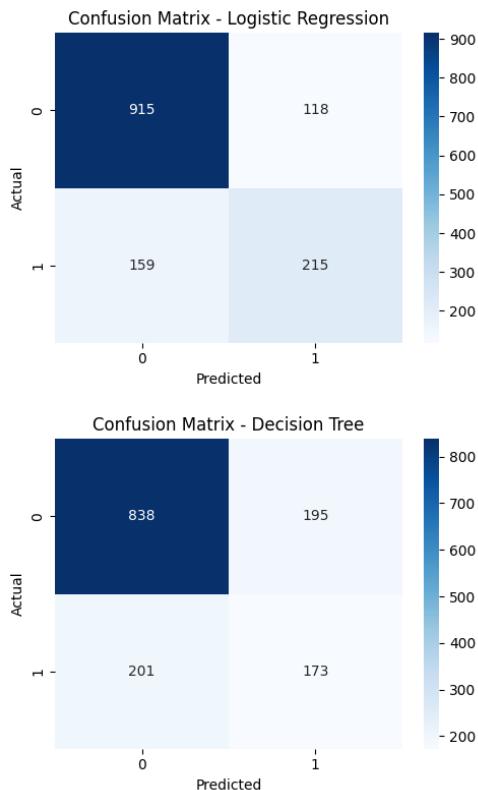


Fig. 1: Confusion Matrices for Logistic Regression and Decision Tree (before tuning).

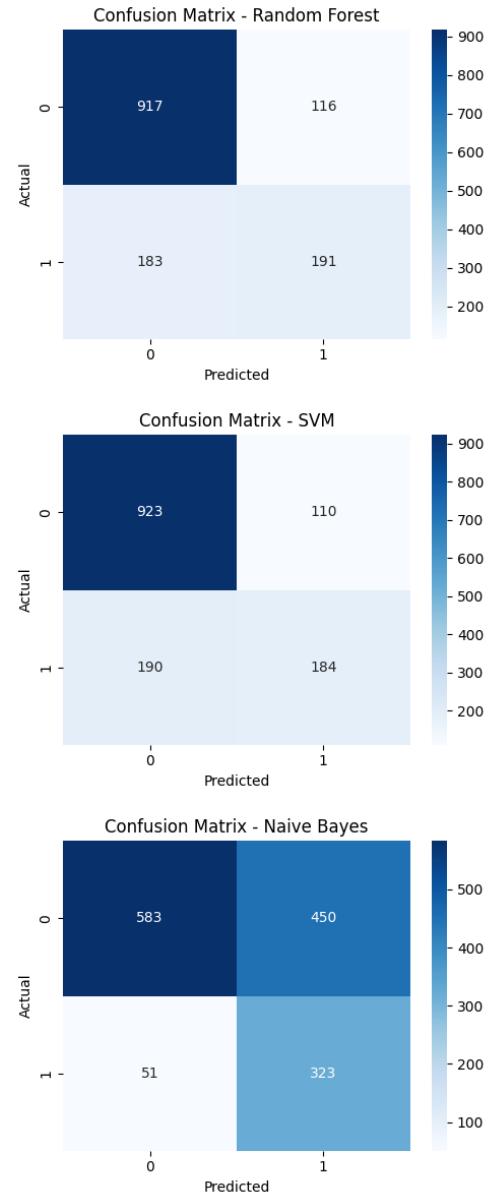


Fig. 2: Confusion Matrices for Random Forest, SVM, and Naive Bayes (before tuning).

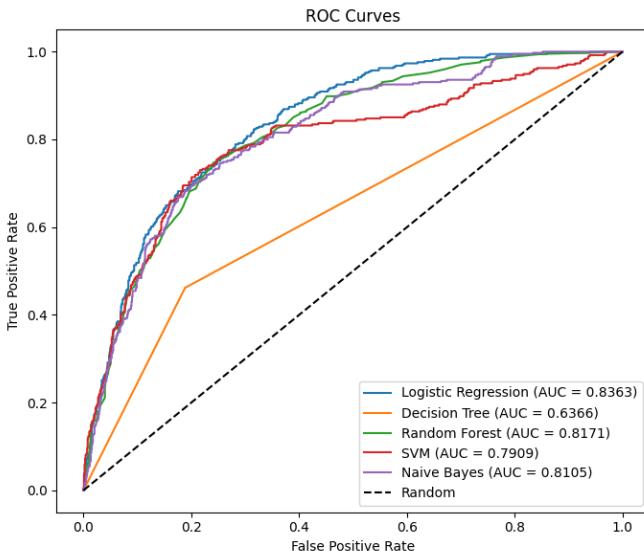


Fig. 3: ROC Curves for all models (before tuning).

### C. Model Optimization and Hyperparameter Tuning

Model optimization was performed using hyperparameter tuning with GridSearchCV, where different parameter combinations were tested for each algorithm to improve accuracy, reduce error rates, and enhance overall model performance.

### D. Model Comparison After Tuning

**Logistic Regression:** It still remains the best overall model after tuning, achieving the highest accuracy among all models and the best ROC–AUC. Precision and recall remain well-balanced, and after tuning we observe slightly improved stability in the F1-score. Logistic Regression continues to be the most reliable baseline with the best overall balance of business-impact metrics.

**Support Vector Machine (SVM):** It showed improvement in precision and F1-score but a slight drop in AUC. After tuning, we see better precision (fewer false positives), better F1-score (improved balance between precision and recall), and a slight accuracy increase. The SVM tuning of C, gamma, and kernel parameters enhanced precision and stability. However, SVM still cannot outperform Logistic Regression because the dataset does not exhibit highly complex non-linear boundaries that strongly benefit from kernel transformations.

**Random Forest:** After tuning, Random Forest achieved higher precision but experienced a small decline in recall. It now has the highest precision of all models, but recall dropped, indicating that the model still misses several churners. The F1-score remains moderate, showing that while the ensemble is stable, further imbalance correction or class weighting might help.

**Decision Tree:** After tuning, Decision Tree demonstrates significant stability improvement. Both precision and recall improved, but it remains comparatively weak overall. The results clearly show that pruning via `max_depth` and

`min_samples_split` substantially improved its generalization and reduced overfitting.

**Naive Bayes:** Even after tuning, Naive Bayes exhibits minimal performance improvement. It maintains extremely high recall but still very low precision, leading to low overall accuracy. No major AUC gains are observed, confirming that the probabilistic assumptions limit its performance on this dataset.

### Summary Observations:

- Hyperparameter tuning increased model stability rather than drastically boosting scores.
- Tree-based models (Random Forest, Decision Tree) would benefit from stronger imbalance handling and threshold calibration.
- Linear decision boundaries perform best, as confirmed by Logistic Regression remaining the top model even after tuning.
- High recall remains challenging across all models, indicating the inherent difficulty of identifying churners.

TABLE II: Model Performance (After Tuning)

Model	Acc.	Prec.	Recall	F1	AUC
Logistic Regression	0.8031	0.6465	0.5722	0.6071	0.8363
SVM (RBF)	0.7989	0.6409	0.5535	0.5940	0.8273
Random Forest	0.7953	0.6792	0.4358	0.5309	0.8369
Decision Tree	0.7754	0.5980	0.4733	0.5284	0.8164
Naive Bayes	0.6496	0.4224	0.8663	0.5679	0.8126

#### E. Confusion Matrices (After Tuning)

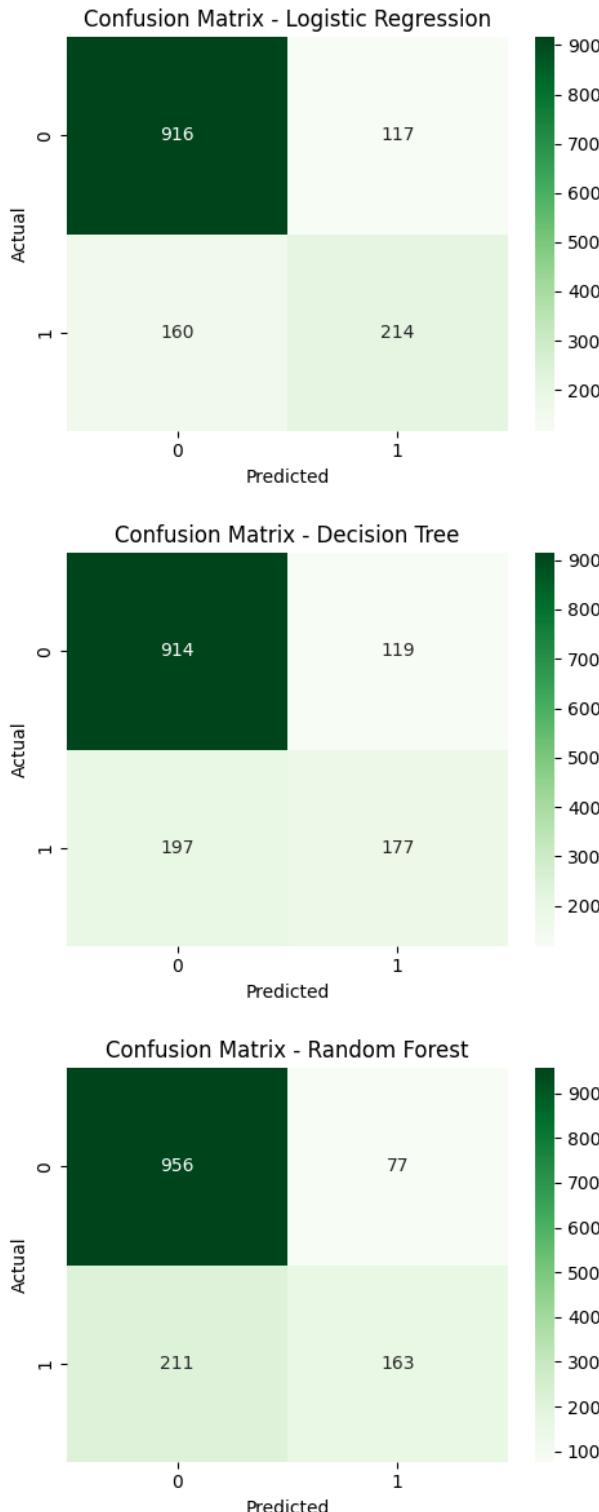


Fig. 4: Confusion Matrices for Logistic Regression, Decision Tree and Random Forest (after tuning).

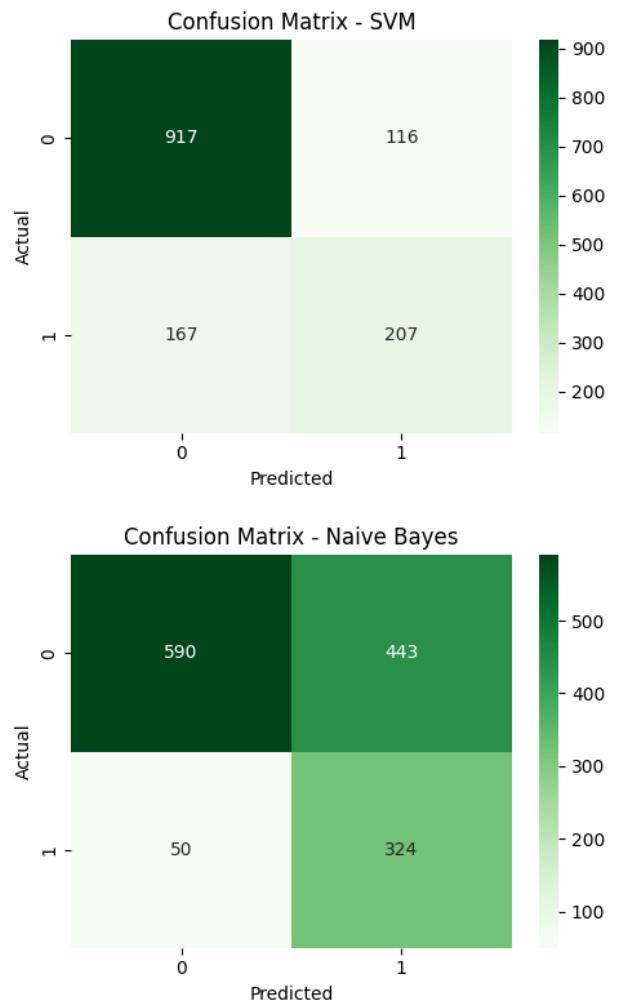


Fig. 5: Confusion Matrices for SVM, and Naive Bayes (after tuning).

#### F. ROC Curves and AUC Analysis

##### Before Tuning:

- Logistic Regression:** Achieved the highest AUC ( $\approx 0.836$ ), showing strong ability to distinguish between churn and non-churn customers.
- Random Forest and Naive Bayes:** Performed well with AUC values around 0.81, indicating good discrimination capability.
- Support Vector Machine (SVM):** Followed closely with an AUC of approximately 0.79, reflecting solid but slightly weaker separation.
- Decision Tree:** Displayed the weakest discrimination power with AUC around 0.63, indicating a higher rate of misclassification.

##### After Tuning:

- AUC values improved slightly for most models, indicating better class separation and calibration.
- Logistic Regression:** Remained the best overall performer with an AUC of approximately 0.836, maintaining its top performance.

tier stability.

- **Random Forest and SVM:** Both improved, approaching or exceeding AUC values of 0.83, reflecting refined separation between churn and retention cases.
- **Decision Tree:** Showed significant improvement (AUC  $\approx 0.81$  compared to  $\approx 0.63$  earlier), becoming far more reliable after tuning.
- **Naive Bayes:** Improved slightly but continued to remain in the mid-performance range, with limited discrimination improvements.

#### Overall Observations:

- Most models exhibited stronger ROC curves and higher AUC after tuning, reflecting enhanced predictive strength.
- **Decision Tree** benefited the most from tuning, showing a major AUC increase.
- **Logistic Regression and Random Forest** consistently remained the top-performing models both before and after tuning, confirming their robustness and generalization strength.

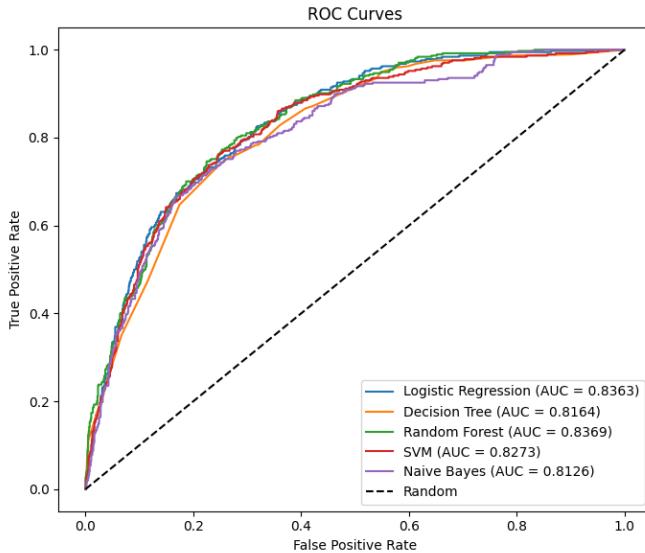


Fig. 6: ROC Curves for all models (after hyperparameter tuning).

#### G. Performance Summary

##### Best Performing Model (Both Before & After):

- **Logistic Regression:** Consistent, stable, and the highest accuracy overall across both pre- and post-tuning phases. It demonstrated strong reliability and minimal performance variance, making it an ideal baseline for churn prediction.

##### Models with Most Improvement After Tuning:

- **Decision Tree:** Exhibited the largest performance jump, particularly in recall and overall stability after pruning and depth optimization.
- **SVM and Random Forest:** Achieved moderate gains in accuracy and reduced classification error through hyperparameter optimization, enhancing precision and generalization.

##### Least Impact of Tuning:

- **Naive Bayes:** Showed minimal improvement due to its strong underlying independence assumptions and lack of tunable parameters.
- **Logistic Regression:** Remained almost unchanged, as it was already near-optimal before tuning, with stable coefficients and robust regularization.

**Overall Observations:** Hyperparameter tuning improved the performance of nearly all models except Logistic Regression, which was already well-optimized. The tuning process proved especially beneficial for complex, high-variance models such as Decision Trees, SVM, and Random Forest, resulting in better predictive accuracy, reduced overfitting, and improved model reliability.

TABLE III: Performance Comparison (Before and After Tuning) – Key Metrics

Model	Accuracy (Before)	Accuracy (After)	Error Rate (Before)	Error Rate (After)
Logistic Regression	0.8031	0.8031	0.1969	0.1969
Random Forest	0.7875	0.7953	0.2125	0.2047
SVM (RBF)	0.7868	0.7989	0.2132	0.2011
Decision Tree	0.7186	0.7754	0.2815	0.2246
Naive Bayes	0.6439	0.6496	0.3561	0.3504

#### H. Business Interpretation

- Customers on month-to-month contracts are more likely to churn.
- High monthly charges correspond to elevated churn risk.
- Longer-term clients display a strong retention probability.
- Models can automate early-warning churn alerts for targeted retention campaigns.

#### VI. Conclusion

Overall, hyperparameter tuning improved the performance of every model except Logistic Regression, which remained almost unchanged. The tuning process was especially helpful for complex, high-variance models like Decision Trees, SVM, and Random Forest, leading to better predictive accuracy and lower error rates.

#### Acknowledgments

We thank the DA 2270 teaching team for their guidance and feedback.

#### References

- [1] M. Tyagi *et al.*, “Customer Churn Prediction Code and Reports,” GitHub Repository, 2025. <https://github.com/monikatyagiisc/customer-churn-prediction>
- [2] Kaggle, “Telco Customer Churn,” <https://www.kaggle.com/datasets/blastchar/telco-customer-churn>.
- [3] Kaggle, “Telco Customer Churn IBM Dataset,” <https://www.kaggle.com/datasets/yeanzc/telco-customer-churn-ibm-dataset>.
- [4] Springer, *Recent Advances in Customer Churn Modeling*, 2024.
- [5] S. Verma and H. Kumar, “A Comparative Study of Machine Learning Techniques for Customer Retention,” *International Journal of Data Science*, vol. 12, pp. 102–115, 2024.