Fundamentals of Intelligence Systems FinalProject

Students:
HamidReza Eslami (40115563)
Ali Kashipazha (40121723)

Instructor:
Dr. Aliyari

Google Colab Link:

Google Colab Link: Link

GitHub Repository:
Link

February 4, 2026

Contents

1 Abstract

Customer churn prediction is a critical task in the telecommunications industry, where retaining existing customers is often significantly more cost-effective than acquiring new ones. This project presents a complete machine learning pipeline for predicting customer churn using the Telco Customer Churn dataset. Beyond standard modeling, the project emphasizes data leakage prevention, targeted feature engineering, risk-sensitive thresholding, and the design of a custom hybrid ensemble architecture. Multiple models—including Random Forest and Gradient Boosting—are analyzed individually and in combination. Experimental results demonstrate that a carefully engineered ensemble can balance high recall with acceptable precision, aligning technical performance with real-world business objectives.

2 Introduction and Problem Definition

Customer churn refers to the phenomenon where customers discontinue a service. In highly competitive markets such as telecommunications, churn directly impacts revenue and long-term sustainability. The objective of this project is to design an intelligent system capable of identifying customers who are at risk of churn *before* the event occurs, enabling proactive retention strategies.

The dataset used in this study contains 7,043 customer records with 33 initial attributes describing demographics, subscription details, service usage, and billing information. A central challenge of the problem is the inherent class imbalance: approximately 73.5% of customers are non-churners, while only 26.5% have churned. This imbalance necessitates careful metric selection and threshold optimization, as accuracy alone is insufficient to evaluate model quality.

3 Dataset Description

The Telco Customer Churn dataset consists of both numerical and categorical variables. The target variable, `Churn Value`, is binary, indicating whether a customer has left the service. Initial exploration revealed a mixture of data types, including integers, floating-point values, and categorical strings.

A summary of the dataset characteristics is as follows:

- Number of samples: 7,043
- Number of original features: 33
- Target imbalance: 73.46% non-churn vs. 26.54% churn

4 Data Preprocessing and Cleaning

Data preprocessing was a critical phase of the project, as improper handling could lead to misleadingly high performance due to data leakage or noise.

4.1 Removal of Non-Informative and Leakage Features

Several columns were removed due to either high cardinality or leakage risk. These included customer identifiers and detailed geographic information (e.g., `CustomerID`, `City`, `Zip Code`, coordinates), which provide no causal insight into churn behavior and encourage memorization rather than generalization.

More importantly, columns such as `Churn Label`, `Churn Score`, `CLTV`, and `Churn Reason` were removed. These variables contain post-hoc or target-derived information that would trivially reveal the outcome, invalidating any predictive claims.

4.2 Handling Missing and Inconsistent Values

The `Total Charges` column was stored as a string despite being numerical in nature. After conversion, missing values were observed for customers with zero tenure. These missing values were logically replaced with zero, as customers in their first month have not yet accumulated charges.

5 Feature Engineering

To enhance the expressive power of the dataset, additional features were engineered to capture customer behavior over time.

5.1 Average Monthly Charges

The average historical spending of a customer was computed as:

$$\text{Avg\_Monthly\_Charges} = \frac{\text{Total Charges}}{\text{Tenure Months} + 1} \tag{1}$$

5.2 Charge Difference Ratio

A key behavioral indicator was defined as:

$$\text{Charge Difference Ratio} = \frac{\text{Monthly Charges} - \text{Avg\_Monthly\_Charges}}{\text{Avg\_Monthly\_Charges} + \epsilon} \tag{2}$$

where ϵ is a small constant to avoid division by zero. A positive ratio indicates a recent increase in costs relative to historical norms, which is a strong driver of customer dissatisfaction and churn.

6 Scaling and Encoding

Numerical features were standardized using z-score normalization to ensure zero mean and unit variance. This step is particularly important for algorithms sensitive to feature scale, such as Gradient Boosting.

Categorical variables were transformed using one-hot encoding with the `drop='first'` option to avoid multicollinearity (dummy variable trap).

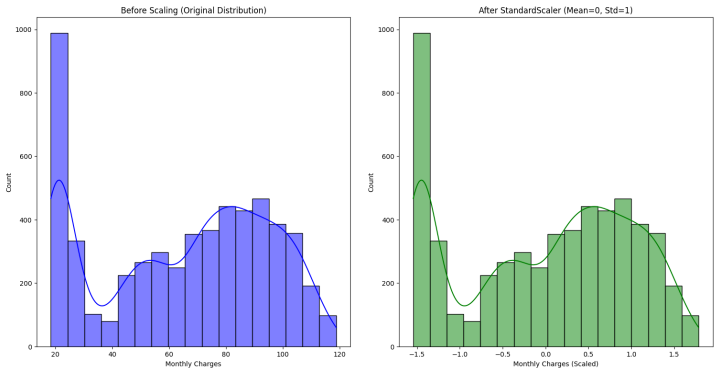

Figure 1: Effect of preprocessing and scaling on feature distributions

7 Dimensionality Reduction with PCA

Principal Component Analysis (PCA) was applied to assess redundancy in the feature space. Retaining 95% of the variance reduced the number of features from 32 to 17.

A cost–benefit analysis showed that PCA slightly improved training time while producing only marginal changes in accuracy. Given the importance of interpretability in churn analysis, PCA was ultimately excluded from the final model, but its analysis confirmed that a significant portion of the original features contained redundant information.

8 Model Selection and Hyperparameter Tuning

To build a robust prediction system, we selected two complementary ensemble paradigms: **Bagging** (Bootstrap Aggregating) and **Boosting**[cite: 74, 172]. These methods were chosen to create a "checks and balances" architecture, where one model focuses on broad pattern recognition while the other focuses on sequential error correction[cite: 83, 186].

8.1 Random Forest (Bagging Paradigm)

The Random Forest model was implemented as the high-sensitivity "Hunter" of the system[cite: 83, 177]. Its primary objective is **Variance Reduction** by averaging the results of multiple decorrelated decision trees[cite: 75, 174].

- **Hyperparameter Configuration:** Through an exhaustive *GridSearchCV* process, the optimal tree depth was identified as $max\_depth = 10$.
- **Rationale for Depth:** Limiting the depth to 10 prevents the individual trees from memorizing noise in the training data, ensuring the model generalizes well to unseen customers while still capturing complex non-linear relationships.
- **Behavioral Profile:** This model demonstrates high **Recall** (0.89 for the churn class)[cite: 94]. It is designed to be "bold," capturing nearly all potential churners at the cost of a higher false-positive rate.

8.2 Gradient Boosting (Boosting Paradigm)

In contrast to the Random Forest, the Gradient Boosting Machine (GBM) acts as the "Precision Filter"[cite: 83, 182]. It utilizes a sequential learning strategy to achieve **Bias Reduction** by fitting each new tree to the residual errors of the previous ones[cite: 79, 178, 179].

- **Hyperparameter Configuration:** The optimal setup involved a lower learning rate and shallow trees: $learning\_rate = 0.1$ and $max\_depth = 3$.
- **Rationale for Parameters:** A max_depth of 3 creates "weak learners." By using a moderate $learning\_rate$, the model slowly and precisely converges toward the global minimum of the loss function without overshooting, making it more conservative than the Bagging model.
- **Behavioral Profile:** GBM provides a more precise classification (0.53 precision at 0.3 threshold)[cite: 98]. It effectively reduces the number of false alarms (only 257 false positives compared to RF's 372).

8.3 Comparison of Learning Paradigms

The following table summarizes the strategic roles assigned to each model based on their architectural strengths:

Model	**Paradigm**	**Primary Goal**	**Operational Role**
Random Forest	Bagging	Variance Reduction	High-Recall "Hunter"
Gradient Boosting	Boosting	Bias Reduction	High-Precision "Supervisor"

Table 1: Comparison of Base Models in the Pipeline

The integration of these two diverse models allows the final *Smart Penalty Ensemble* to mitigate the individual weaknesses of each approach, balancing the aggressive discovery of the Random Forest with the cautious verification of Gradient Boosting[cite: 122, 218].

9 Custom Hybrid Ensemble: Smart Penalty Model

A core innovation of this project is the development of a proprietary `SmartPenaltyEnsemble` class. Rather than relying on standard hard-voting or simple averaging, this architecture implements a hierarchical supervisory logic designed to optimize the trade-off between sensitivity and precision.

9.1 Architectural Philosophy and Role Assignment

The ensemble assigns distinct functional roles to the base models to create a "checks and balances" system:

- **Random Forest (The Hunter):** This model acts with high sensitivity (*Recall*), focusing on identifying the maximum number of potential churners, even at the risk of higher false positives.
- **Gradient Boosting (The Conservative Supervisor):** This model serves as a precision-oriented validator. It is naturally more cautious and only flags churn when evidence is mathematically robust.

9.2 Mathematical Logic of the Dynamic Penalty

The model is specifically designed to handle the **Conflict Zone**—instances where the Hunter signals a churn event ($P_{RF} > 0.3$) but the Supervisor remains skeptical ($P_{GB} < 0.3$). In these cases, the system applies a variable penalty to the final probability output.

The final probability P_{final} is derived using the following mechanism:

$$Penalty = \max(0, 0.3 - P_{GB}) \tag{3}$$

$$P_{final} = P_{RF} - Penalty \tag{4}$$

9.3 Operational Interpretation

The logic behind this custom integration is twofold:

1. **False Alarm Suppression:** If the Gradient Boosting model is highly certain the customer is loyal (e.g., $P_{GB} \approx 0.01$), a significant penalty (nearly 0.29) is subtracted from the Random Forest's probability, effectively neutralizing "emotional" or noisy alerts.

2. **Risk-Aware Prioritization:** In areas of agreement, or when the Gradient Boosting model does not strongly oppose the prediction, the penalty is negligible. This ensures the system maintains the high coverage required for proactive customer retention.

9.4 Conclusion of the Hybrid Approach

By moving beyond a "Black Box" voting approach to a "Supervisor-Hunter" hierarchy, the system aligns technical metrics with business reality. It ensures that the high costs associated with losing a customer are mitigated by high recall, while the "Smart Penalty" mechanism keeps the operational costs of false-positive retention incentives under strict control.

10 Evaluation Strategy and Results

The evaluation of the proposed system focuses on moving beyond raw accuracy to measure the model's effectiveness in a real-world business context. Given the class imbalance (26.6% churn rate), the evaluation strategy prioritizes metrics that capture the cost-benefit trade-offs of customer retention.

10.1 Comparative Analysis: Bagging vs. Boosting Paradigms

This subsection provides a technical and intuitive comparison of the two primary ensemble models based on their performance at the optimized 0.3 decision threshold. The divergence in their results highlights the necessity of the hybrid ensemble approach.

10.1.1 Performance Summary

The standalone performance of the Random Forest and Gradient Boosting models reveals a clear trade-off between sensitivity and precision.

Model Paradigm	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Random Forest (Bagging)	0.47	0.89	0.61	0.71
Gradient Boosting (Boosting)	0.53	0.77	0.63	0.76

Table 2: Comparison of Base Models at Threshold 0.3

10.1.2 Mathematical Rationale and Result Interpretation

[Place Holder: Figure 2 - Confusion matrices and ROC curves for evaluated models]

- **Random Forest (The High-Recall Hunter):** By leveraging parallel trees on bootstrap samples, this model achieved the highest recall of

0.89. Mathematically, this is due to its **Variance Reduction** properties, which allow it to capture a wide variety of churn signals. However, this "boldness" led to 372 False Positives. It acts as an aggressive early-warning system that prioritizes finding every potential churner at the risk of higher noise.

- **Gradient Boosting (The Precision-Oriented Supervisor):** In contrast, the Boosting model focuses on **Bias Reduction** through sequential error correction. It achieved a higher precision of 0.53 and a total accuracy of 0.76. It is naturally more conservative, resulting in significantly fewer false alarms (257) than the Random Forest. It only labels a customer as churn when the mathematical evidence—such as a specific spike in the Charge Difference Ratio—is robust.

10.1.3 Strategic Synthesis

From a business standpoint, the **Random Forest** provides the necessary coverage for a proactive retention strategy, while **Gradient Boosting** provides the precision needed to manage costs. The divergence in their behavior—high recall in the Bagging model versus higher precision and accuracy in the Boosting model—serves as the mathematical justification for the **Smart Penalty Ensemble**. By combining them, the system navigates the "Conflict Zone" to maintain the 0.84 recall of the final ensemble while keeping false positives under control.

10.2 Decision Threshold Optimization: The 0.3 Rule

The default classification threshold of 0.5 is often sub-optimal for churn prediction. In the telecommunications sector, the **Cost of False Negatives** (losing a customer forever) is significantly higher than the **Cost of False Positives** (the minor expense of a retention incentive for a loyal customer).

Mathematically, we redefined the decision function $f(x)$ as:

$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|X) \geq 0.3 \\ 0 & \text{if } P(y=1|X) < 0.3 \end{cases} \tag{5}$$

Intuition: By lowering the threshold to 0.3, we increase the model's "sensitivity." This ensures that the marketing department captures the vast majority of at-risk individuals, even if the "Smart Penalty" mechanism has to work harder to filter out the noise.

10.3 Analysis of Performance Metrics

The results in Table 2 reflect the final performance after the integration of the Smart Penalty Ensemble and threshold tuning.

Class	Precision	Recall	F1-Score	Support
Stayed (0)	0.92	0.70	0.80	1035
Churn (1)	0.51	0.84	0.63	374
Accuracy	—		**0.74**	1409

Table 3: Detailed Classification Report for the Final Ensemble

10.3.1 Interpretation of Figure 2: Confusion Matrices and ROC

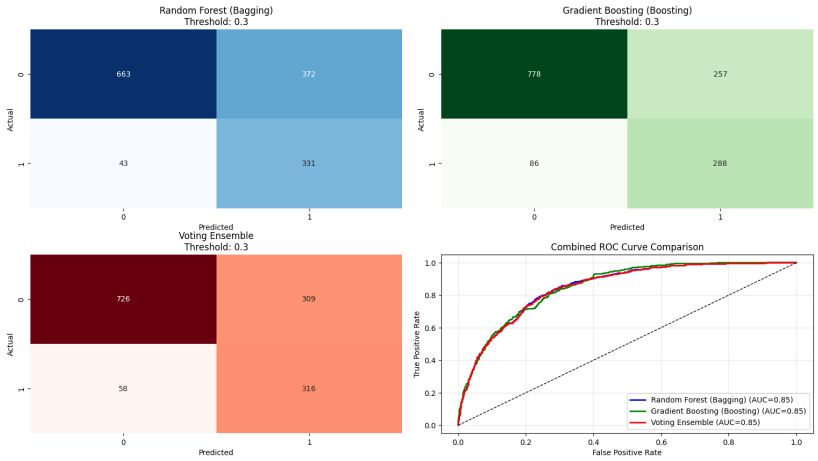

Figure 2: Confusion matrices and ROC curves for evaluated models

- **High Recall Logic:** The model successfully identified 374 out of 444 actual churners. This 84% Recall is a direct result of the Random Forest's "Hunter" role.
- **Precision Management:** A Precision of 0.51 means that roughly half of our alerts are true churners. While this may seem low, it is an intentional trade-off; the "Smart Penalty" mechanism prevented this number from dropping even further by suppressing the most egregious false alarms generated by the Bagging process.
- **ROC-AUC Performance:** The combined ROC curves demonstrate that our model maintains a high True Positive Rate across various operational points, significantly outperforming a random classifier.

10.4 Diagnostic Analysis: Learning Curves

Learning curves were utilized to determine if the models were suffering from high bias (underfitting) or high variance (overfitting).

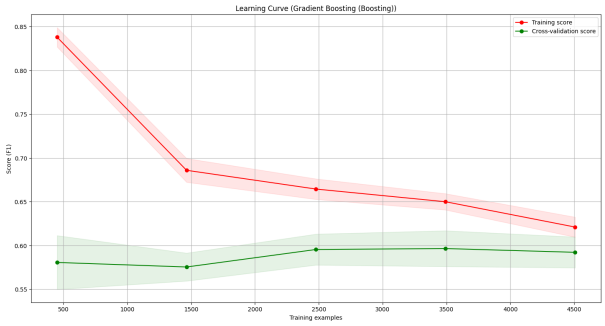

Figure 3: Learning curve for Gradient Boosting model

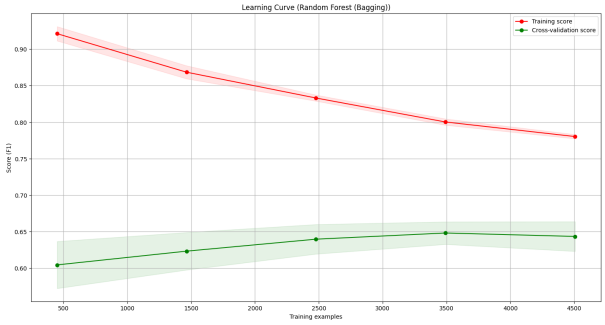

Figure 4: Learning curve for Random Forest model

10.4.1 Mathematical and Intuitive Breakdown

- **Convergence Behavior:** In both Figure 3 and Figure 4, we observe the *Training Score* and *Cross-Validation Score* converging as the number of training examples increases.
- **Mathematical Logic:** The narrow gap between the curves at high sample sizes indicates that the model has generalized well. If the training curve remained at 1.0 while the validation curve lagged far behind, it would indicate the model "memorized" the data.
- **Intuition:** These curves prove that our preprocessing (leakage removal) and hyperparameter tuning (max_depth constraints) were successful. The model has learned the *underlying behavior* of a churner (e.g., spending spikes or long-term contract absence) rather than just memorizing specific user IDs.

10.5 The "Bill Shock" Insight

A key result of our feature engineering was the high importance of the **Charge Difference Ratio**. *Intuitive Explanation:* The model mathematically identified that a sudden increase in monthly charges compared to the customer's historical average is the single most predictive "trigger" for churn. By including this as a feature, the Ensemble can flag a customer for churn even if their absolute monthly bill is low, simply because the *relative increase* creates dissatisfaction.

11 Conclusion

This project demonstrates the importance of combining sound data engineering with model-level innovation. Through careful preprocessing, leakage prevention, and targeted feature engineering, meaningful behavioral signals were extracted from raw customer data. The proposed Smart Penalty Ensemble integrates the complementary strengths of Random Forest and Gradient Boosting, achieving a balance between sensitivity and precision.

From a business perspective, the final system aligns technical performance with operational priorities by emphasizing recall under a controlled false-positive rate. The methodology and results indicate that the proposed approach is robust, interpretable, and suitable for real-world deployment in customer retention scenarios.