Final Project Proposal:
Customer Churn Prediction Using Machine Learning

Jesigga Sigurdardottir, Cahide Tuncer, Annie Phan

February 4, 2026

Abstract

Customer churn is an important business problem that affects revenue and growth. This project shows the development of a machine learning model to predict customer churn based on historical behavioral, transactional, and demographic data. By identifying customers who are likely to leave, businesses can take proactive steps to retain them. Our approach tests multiple classification algorithms for churn prediction, including Logistic Regression, Naive Bayes, k-Nearest Neighbors, Decision Tree, Random Forest, AdaBoost, XGBoost, and Neural Networks. The goal is to build an accurate churn prediction system.

1 Introduction

Maintaining customers is often less expensive than gaining new customers, making churn prediction a critical objective for data-driven organizations. With the growing availability of customer data, supervised machine learning techniques offer powerful tools to identify patterns associated with churn behavior. This project focuses on comparing multiple supervised learning models to predict whether a customer will discontinue a service. This allows companies to implement ways to reduce customer churn. Datasets can vary in noise and consistency; relying on a single model may not yield consistent results across different datasets. Therefore, this study evaluates several supervised learning approaches to determine the best-performing model for the given dataset. Organizations seeking to identify the most effective churn prediction strategy for their own data can adopt a similar comparative modeling framework.

2 Problem Statement

Given customer data as input, the goal of this project is to predict whether a customer will leave a business after a certain period of time. The objective of this project is to predict customer churn using historical customer data. Customer churn is the probability that a customer will leave a business. It is predicted based on demographic information, past browsing information, historical purchase information, identification information, and customer feedback information. This is a binary classification problem, where the churn variable indicates whether or not a customer has churned. Key challenges in this task include effective feature selection and model interpretability.

To address these challenges, the project aims to evaluate multiple supervised classification models to determine the most accurate approach to predict customer churn. Different models may emphasize different features and assign varying levels of importance to each, providing insight into which customer characteristics are most influential in churn prediction.

3 Related Work

3.1 Machine Learning: Algorithms, Real-World Applications and Research Directions[1]

This paper is an overview of machine learning algorithms. Rather than focusing on a specific dataset or case, such as churn prediction, it serves as an overall reference for understanding core machine learning concepts and their application across domains. The paper discusses different types of data used in their application, containing structured, unstructured, semi-structured data, and metadata, and explains how data characteristics influence model selection.

This includes supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning, and explains when to use each learning process. Different learning objectives, such as classification, clustering, association, and dimensionality reduction, are explained, with examples of when each is used. Many types of classification are mentioned, including binary, multi-class, and multi-label classification.

The paper also covers regression, including simple and multiple linear regression, polynomial regression, and regularization methods such as Lasso and ridge regression, while explaining their advantages and disadvantages. Unsupervised learning methods are explained through clustering, describing several widely used clustering techniques. Dimensionality reduction and feature learning methods are discussed through a range of basic feature selection and extraction approaches to more statistical techniques, such as ANOVA and chi-squared tests.

Association rule learning is mentioned, with details on the Apriori algorithm and its role in discovering relationships within a large dataset. Reinforcement learning is decision making through reward based feedback. The paper also discusses artificial neural networks and deep learning, including multilayer perceptrons, convolutional neural networks, and long short-term memory recurrent neural networks, which are the most commonly used deep learning approaches.

The paper ends with machine learning applications, covering business analytics, medical image based cancer detection, and livestock management. This demonstrates the versatility and growing impact of machine learning across multiple industries.

3.2 A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector[2]

This paper focuses on developing a customer churn prediction model for the telecommunications industry using machine learning. They explore different machine learning methods to identify the customers who are most likely to end their service. This paper specifically uses Random Forest as the predictive method.

The paper begins with reviewing challenges with prediction, including class imbalance in telecommunication datasets and the difficulty of extracting meaningful patterns from a large amount of customer behavior data. To address challenges, the Random Forest algorithm is used to analyze the importance of variables to identify factors that contribute to customer churn, or customers ending their service.

The paper shows how Random Forest can be used for churn prediction while also showing which customer variables are associated with churn risk. Overall, the paper was created to help the Telecom Sector better understand customer churn behavior and implement predictive models to support keeping previous customers.

4 Proposed Solution

4.1 Workflow

The proposed solution tackles the customer churn prediction task by developing, training, and systematically comparing multiple supervised learning models to identify the approach that achieves the strongest predictive performance. To ensure rigor and reproducibility, the study follows a structured machine learning pipeline consisting of five key stages: **data cleaning and preprocessing, exploratory data analysis (EDA), feature engineering and selection, model training with hyperparameter tuning, and comprehensive model evaluation and interpretation.** The overall workflow is summarized in Figure 1, which illustrates the end-to-end process used to develop, optimize, and evaluate churn prediction models.

Data cleaning and preprocessing: The data cleaning and preprocessing stage begins with a systematic review of the dataset, including its structure, feature types, and churn target distribution, to evaluate data quality and detect potential class imbalance, as well as to assess whether additional handling is needed during model development. This stage facilitates the identification of common data issues such as missing values, duplicated records, and outliers that could negatively impact predictive performance. Missing values are handled using feature-appropriate imputation strategies, including mode imputation for categorical variables and mean or median imputation for numerical variables. Duplicate entries and invalid observations are removed to improve data consistency and reliability. Outliers are detected using statistical methods and either treated or excluded to reduce their disproportionate influence on model training. Categorical features are then encoded into numerical representations (e.g., one-hot encoding for nominal variables and label encoding for ordinal variables) to ensure compatibility with machine learning algorithms. Numerical features are scaled where appropriate to support comparable feature ranges across models. Finally, the dataset is partitioned into training, validation, and test sets to enable unbiased evaluation and support robust assessment of model generalization performance.

Exploratory data analysis (EDA): Exploratory data analysis is conducted primarily on the training set to examine patterns and relationships relevant to churn prediction while avoiding information leakage into the validation and test sets. The analysis starts by assessing the proportion of churned versus retained customers to quantify class imbalance. Feature distributions are

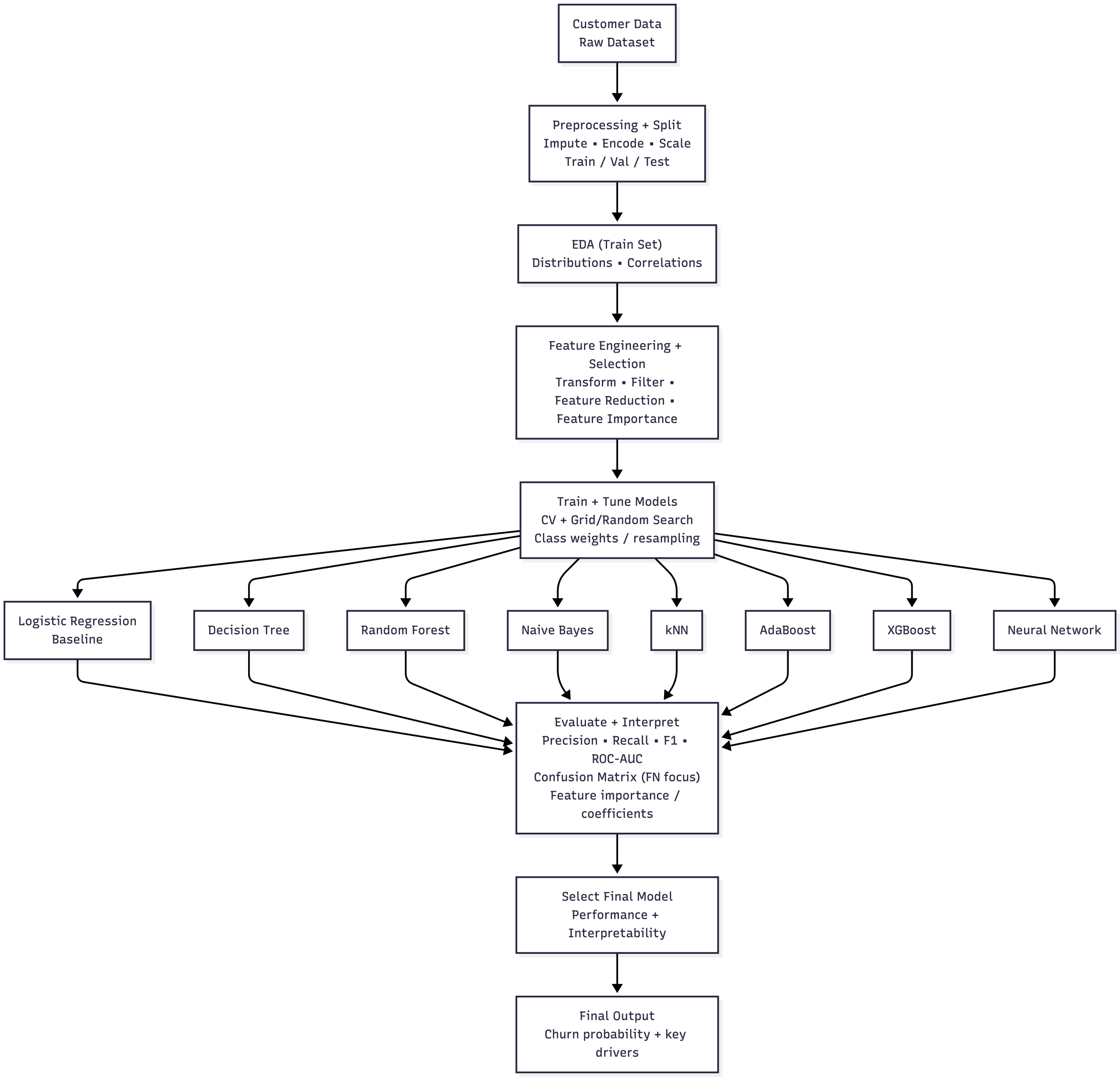
Customer Data
Raw Dataset
Preprocessing + Split
Impute ▪ Encode ▪ Scale
Train / Val / Test
EDA (Train Set)
Distributions ▪ Correlations
Feature Engineering + Selection
Transform ▪ Filter ▪ Feature Reduction ▪ Feature Importance
Train + Tune Models
CV + Grid/Random Search
Class weights / resampling
Logistic Regression Baseline
Decision Tree
Random Forest
Naive Bayes
kNN
AdaBoost
XGBoost
Neural Network
Evaluate + Interpret
Precision ▪ Recall ▪ F1 ▪ ROC-AUC
Confusion Matrix (FN focus)
Feature importance / coefficients
Select Final Model
Performance + Interpretability
Final Output
Churn probability + key drivers

Figure 1: Machine Learning Pipeline for Customer Churn Prediction

explored using descriptive statistics and visualizations to characterize customer behavior. Comparative analyses are performed to highlight systematic differences in key behavioral attributes, such as customer engagement, activity recency, interaction frequency, and transaction value, between churned and retained customers. In addition, correlation analysis is used to evaluate relationships among predictor variables and their association with churn, as well as to detect potential multicollinearity. Findings from this stage guide subsequent feature engineering and model selection decisions.

Feature engineering and selection: Feature engineering and selection focus constructing informative predictors from raw data using domain knowledge and selecting the most relevant features to improve model performance. Guided by EDA findings, additional features are derived to better capture customer engagement patterns and lifecycle behavior. Feature transformations are applied when needed to address skewed distributions and support more effective model learning. Feature selection is performed using complementary approaches, including correlation-based filtering to remove highly correlated and redundant predictors, as well as model-based importance analysis using a random forest to identify high impact variables. This stage aims to balance predictive performance with interpretability, ensuring that the final feature set supports robust and explainable churn prediction.

Model training with hyperparameter tuning: Multiple classification algorithms are trained on the training set to predict customer churn. Logistic regression serves as a baseline model to establish a reference level of performance due to its simplicity and interpretability. More advanced models, including Naive Bayes, k-Nearest Neighbors (kNN), Decision Tree, Random Forest, AdaBoost, XGBoost, and neural networks, are subsequently explored to capture nonlinear patterns and complex feature interactions. Hyperparameters are tuned using systematic search strategies (e.g., grid search or randomized search) with cross-validation to improve generalization and enable fair comparison across models. Class imbalance is addressed during training through techniques such as class weighting or resampling to reduce bias toward the majority non-churn class and improve churn detection. The best hyperparameter settings are selected based on validation performance using evaluation metrics appropriate for churn prediction, and the top-performing models are then assessed on the test set.

Model evaluation and interpretation: Model performance is evaluated to assess predictive reliability, robustness, and practical value for churn prediction. Results are reported on both the validation set and the test set using metrics appropriate for imbalanced classification, including precision, recall, F1-score, and ROC-AUC. Confusion matrices are analyzed to characterize error patterns, with particular emphasis on false negatives corresponding to missed churn cases. To complement quantitative evaluation, interpretability methods are applied to explain model behavior and highlight key drivers of churn, including feature importance for tree-based models and coefficient-based analysis for linear models. The final model is selected by jointly considering predictive performance and interpretability to support actionable and explainable decision-making.

4.2 Model Selection

Customer churn prediction is formulated as a supervised binary classification task, where the response variable indicates whether a customer is expected to churn (1) or remain active (0). The primary objective is to accurately identify customers at high risk of churn to proactive retention strategies. To establish a strong baseline and evaluate whether more complex models provide meaningful performance gains, this study compares multiple classification algorithms for churn prediction, including **Logistic Regression, Naïve Bayes, k-Nearest Neighbors, Decision Tree, Random Forest, AdaBoost, XGBoost, and Neural Networks.**

Logistic Regression: Logistic regression is a widely used supervised learning method for binary classification and is adopted as a baseline model in this study to estimate customer churn probability. Given a customer feature vector, the model computes the probability of churn using a sigmoid function, producing an output between 0 and 1. These probabilities are subsequently converted into class labels using a decision threshold, commonly set at 0.5 but adjustable based on business objectives and recall requirements. A key advantage of logistic regression is its transparency: model coefficients provide direct insight into how each predictor influences churn likelihood, making the results easy to interpret and communicate. Due to its efficiency, stability, and explainability, logistic regression serves as a strong reference point against which more complex models can be evaluated.

Naive Bayes: Naïve Bayes is used as a fast and lightweight probabilistic classifier that serves as an efficient baseline for churn prediction. It is based on Bayes' theorem and estimates the probability of churn by assuming conditional independence among predictor variables given the class label, meaning each feature contributes independently to the prediction. Although this assumption is often unrealistic in real-world customer behavior data, Naïve Bayes can still perform competitively in many classification settings due to its fast training time and robustness in high-dimensional feature spaces. As a result, it is well-suited for benchmarking and rapid model comparison, allowing this study to evaluate whether more complex models provide meaningful performance improvements for churn prediction.

k-Nearest Neighbors (kNN): k-Nearest Neighbors (kNN) is employed as a simple and intuitive distance-based classification method for churn prediction. It assigns a class label to each customer by identifying the k most similar customers in the training data using a chosen distance metric (e.g., Euclidean distance) and predicting the majority class among these neighbors. Because kNN makes no strong assumptions about the underlying data distribution, it can capture nonlinear decision boundaries and local patterns in customer behavior. However, kNN is sensitive to feature scaling and the choice of k, so appropriate preprocessing and hyperparameter tuning are required to ensure reliable performance. This approach serves as a baseline to evaluate whether similarity-based classification can achieve competitive churn prediction performance relative to tree-based and boosting methods.

Decision Tree: Decision Trees are included in this study due to their strong interpretability and ability to capture nonlinear decision boundaries and simple feature interactions. A Decision Tree is a supervised learning model that performs classification through a hierarchical, rule-based structure by partitioning the input feature space using split criteria such as Gini impurity or entropy to separate churned and non-churned customers. The tree starts at a root node and repeatedly splits the dataset into child nodes based on decision rules until the stopping criteria are met, and leaf nodes assign the final class label. Each internal node corresponds to a specific feature and split condition, making the overall decision process transparent and easy to interpret. Despite these advantages, individual Decision Trees can be prone to overfitting and may not capture complex patterns in the data; therefore, they are used as an interpretable baseline and compared with ensemble models in the overall evaluation.

Random Forest: Random Forest is a tree-based ensemble method used for classification that improves predictive stability and generalization compared to a single Decision Tree, while effectively capturing nonlinear relationships in the data. The model constructs multiple decision trees using bootstrap sampling of the training set and introduces additional randomness by selecting a subset of features at each split. Predictions are aggregated across trees, typically using majority voting for classification, to produce a more robust final output. By averaging across many trees, Random

Forest reduces variance and mitigates overfitting, making it well-suited for churn prediction where customer behavior patterns may be complex and noisy. In addition, Random Forest provides feature importance estimates that support interpretability and help identify key factors associated with churn risk.

AdaBoost: AdaBoost (Adaptive Boosting) is examined as a boosting-based ensemble method designed to improve classification performance by combining multiple weak learners into a stronger predictive model. Instead of training models independently, AdaBoost builds learners sequentially and assigns higher weights to observations that are misclassified in earlier iterations, encouraging subsequent learners to focus on harder-to-predict churn cases. Final predictions are obtained by aggregating the weighted outputs of all learners, producing a more accurate and refined decision boundary than a single model. Due to its ability to reduce bias and enhance predictive accuracy, AdaBoost serves as a strong ensemble benchmark for churn prediction. However, because boosting methods can be sensitive to noisy data and outliers, AdaBoost is evaluated alongside other models to ensure robustness and generalization.

XGBoost: XGBoost (Extreme Gradient Boosting) is selected because it is a high-performance boosting algorithm that is widely used for classification on structured and tabular datasets. It builds an ensemble of decision trees sequentially, where each new tree is trained to correct errors from prior trees by minimizing a loss function through gradient-based optimization. Compared to traditional boosting approaches, XGBoost incorporates regularization and efficient training procedures that improve generalization and reduce the risk of overfitting. Its ability to capture complex nonlinear relationships and feature interactions makes it a strong candidate for churn prediction, where customer behavior patterns are often multifaceted. In addition, XGBoost provides feature importance measures that support the interpretation of key churn drivers, enabling both strong predictive performance and actionable insights.

Neural Networks: Neural Networks are evaluated as flexible, high-capacity models capable of learning complex nonlinear relationships and feature interactions for churn prediction. A neural network consists of multiple layers of interconnected nodes, where each layer applies weighted transformations and nonlinear activation functions to progressively learn higher-level representations from the input features. During training, model parameters are optimized using gradient-based learning to minimize a classification loss, allowing the network to capture patterns that may not be well represented by traditional linear or tree-based models. Due to their expressive power, neural networks can be effective for modeling multifaceted customer behavior signals in churn prediction tasks. However, they may require careful tuning and regularization to prevent overfitting and ensure stable generalization, particularly when the dataset is limited or noisy.

5 Proposed List of Experiments

The proposed experiments are designed to systematically evaluate multiple churn prediction models under consistent preprocessing and validation procedures. As summarized in Table 1, the experimental plan is structured to establish baseline performance, quantify improvements from feature engineering, compare model families, and assess the impact of hyperparameter tuning and class imbalance handling. Model performance will be evaluated using metrics appropriate for imbalanced classification, including precision, recall, F1-score, and ROC-AUC, and final results will be reported on the test set to estimate generalization performance. The split percentage is 70% for the training data, 15% for the validation data, and 15% for the test data.

Exp.	Experiment	Data Split	Design	Primary Metrics
1	Baseline (Logistic Regression)	Train (70%) , Validation (15%)	Train LR baseline on cleaned features; establish reference performance.	Precision, Recall, F1, ROC-AUC
2	Feature engineering impact	Train (70%) , Validation (15%)	Compare model performance before vs. after engineered engagement features.	F1, ROC-AUC
3	Model comparison	Train (70%) , Validation (15%)	Compare LR, NB, kNN, DT, RF, AdaBoost, XGBoost, NN under the same pipeline.	F1, ROC-AUC
4	Hyperparameter tuning	Train (cross-validation), Validation (15%)	Grid/Random Search with cross-validation; select best model configurations.	F1, ROC-AUC
5	Class imbalance handling	Train (70%) , Validation (15%)	Compare class weighting vs. over/under-sampling; assess churn detection gain.	Recall, F1
6	Threshold adjustment	Validation (15%)	Tune threshold to optimize business trade-off between precision and recall.	Precision–Recall, F1
7	Interpretation	Test (15%)	Interpret final models using coefficients and feature importance; apply SHAP for global/local explanations; use PDP/ICE plots to visualize key drivers.	–

Table 1: Experimental design plan including dataset usage and evaluation focus.

6 Experiment Setup and Dataset Details

6.1 Dataset Description

In this project, the Customer Churn dataset shared on Kaggle[3] will be used. The Customer Churn dataset is a tabular dataset where each row represents a single customer. Churn rate is a commonly used metric in marketing and refers to the number of customers who leave a company within a certain period of time.

The main goal of this project is to predict whether a customer will leave the business or not based on the features provided in the dataset. The target variable `churn_risk_score` indicates whether a customer exits the system (1) or stays (0).

The dataset consists of a total of 36,992 rows and 24 columns. Among these features, 7 columns contain numerical data, including the customer ID stored in an unnamed column, while the remaining 17 columns are categorical variables.

The features in the dataset can be analyzed under five main categories:

Identification features include information related to user identification and their entry into the platform, and these features are not used directly in the prediction process. This category includes the customer ID (unnamed column), `security_no`, and `joining_date`.

Demographic features describe the demographic profiles of customers and help analyze the data based on different customer groups. These features include `age`, `gender`, `region_category`, `membership_category`, `referral_id`, and `joined_through_referral`.

Browsing behavior features provide information about how customers use the platform and support the analysis of user engagement patterns. This group includes features such as `avg_time_spent`, `avg_frequency_login_days`, `days_since_last_login`, `internet_option`, and `last_visit_time`.

Historical purchase features represent the economic relationship between users and the business and help analyze how customers perceive the value of the platform. These features include `avg_transaction_value`, `points_in_wallet`, `used_special_discount`, `medium_of_operation`, `offer_application_preference`, and `preferred_offer_types`.

Feedback features provide information that helps analyze users' opinions and feedback about the platform. These features include `past_complaint`, `complaint_status`, and `feedback`.

The target variable of the dataset, `churn_risk_score`, does not belong to any of these feature categories.

6.2 Experiment Setup

Problem Formulation: Since the target variable `churn_risk_score` stores binary values (0 or 1) that indicate whether a customer stays with the business or leaves, and the churn risk outcome is predicted as a stay-or-leave decision rather than a numerical value, the prediction task is formulated as a binary classification problem.

Data Splitting: The dataset will be divided into three subsets in order to evaluate the models: training, validation, and test datasets. The training set will be used to train the models and is planned to contain 70% of the dataset. The validation set will be split as 15% of the data and will be used for model selection and hyperparameter tuning. The test set will also consist of 15% of the data and will be reserved for evaluating the final performance of the models.

Since churn datasets are generally prone to class imbalance, the distribution of the target variable was examined. It was observed that the dataset contains approximately 54% of class 1 with 20012 customers and 46% of class 0 with 16980 customers, which indicates a slight class imbalance. Therefore, a stratified splitting strategy is planned to be applied during the training, validation, and test split in order to preserve the class distribution across all subsets.

Data Preprocessing: Before model training, several checks and preprocessing strategies will be applied in order to prepare the data for modeling. First, missing values will be examined. A standard missing value check using the `isnull()` function indicates that three features (`region_category`, `preferred_offer_types`, and `points_in_wallet`) contain `NaN` values, which are used in Python to represent undefined or missing data. However, this does not necessarily mean that the remaining features are fully free of data quality issues. Therefore, data consistency checks will also be conducted to identify potential irregularities. For some columns, values that are not meaningful but are not explicitly marked as NaN will be examined, and such values will be replaced with NaN when necessary. After these checks, missing values will be handled using feature-appropriate imputation strategies. Numerical features are planned to be imputed using mean or median values, while categorical features will be imputed using the mode, which represents the most frequent data. In addition, duplicate records will be checked and removed if detected, in order to improve data consistency.

To prevent data leakage, the entire data preprocessing process will be performed using only the previously defined training dataset. All required preprocessing steps, such as encoding and missing value imputation, will then be consistently applied to the validation and test datasets.

Feature Encoding and Scaling: Some categorical features in the dataset need to be converted into numerical representations in order to be meaningful during the modeling process. For example, the gender feature contains categorical values representing female and male customers (F/M). Since machine learning models cannot directly interpret categorical text values, one-hot encoding will be applied to this feature, converting the F/M categories into binary numerical values (0/1). Similarly, other nominal categorical features that represent different class information and are expected to provide important signals for predicting churn risk rate will also be encoded using one-hot encoding. These features include `region_category`, `joined_through_referral`, `preferred_offer_types`, `medium_of_operation`, `internet_option`, `used_special_discount`, `complaint_status`, `past_complaint`, and `offer_application_preference`.

In addition to nominal categorical variables, the dataset also contains an ordinal categorical feature. The membership category feature represents a clear hierarchical structure (Basic Membership, Silver Membership, Gold Membership, Premium Membership). Due to this natural ordering, label encoding is planned to be applied to this feature.

As previously mentioned, the dataset contains seven numerical columns. Among these, customer ID and `security_no` do not provide meaningful information for the modeling task and will therefore be excluded from model training. For the remaining numerical features, scaling checks will be performed. When necessary, especially for scale-sensitive models, numerical features with different value ranges will be scaled to ensure fair model learning.

Outliers and Feature Engineering: Outliers in the dataset will be identified using statistical methods and will be treated or removed when necessary in order to reduce their potential negative impact on model performance. In addition, if required, feature engineering techniques may be applied by deriving new features from existing ones, and these newly created features will also be included in the modeling process.

Model Evaluation and Hyperparameter Tuning: As discussed in detail in the Model Selection section, this project aims to compare the predictive performance of multiple machine learning models in order to identify the approach that is most suitable for the given dataset. For this purpose, both highly interpretable models and more complex, high-capacity models will be evaluated. The models planned to be used in this study include Logistic Regression, Naive Bayes, k-Nearest Neighbors, Decision Tree, Random Forest, AdaBoost, XGBoost, and Neural Networks. For all models, the same data splitting ratios and strategy will be applied, and the dataset will be prepared using identical preprocessing steps to ensure a fair and consistent comparison.

To enable a reliable comparison across different models, hyperparameter tuning is planned as part of the experimental setup. During the hyperparameter optimization process, grid search and randomized search methods will be applied exclusively on the training dataset, using k-fold cross-validation (e.g., 5-fold cross-validation). This approach allows the generalization performance of the models to be evaluated more reliably, while also reducing the risk of overfitting to a specific subset of the data.

Explainability and Interpretability: As part of the experimental setup, explainability methods will be integrated in order to better understand the prediction processes of the models from an interpretability and explainability perspective. For baseline models, the coefficients of Logistic Regression will be used to understand the direction and magnitude of feature effects. For tree-based models, feature importance analysis will be used to identify which features are most influential in churn prediction. At the same time, SHAP (SHapley Additive exPlanations) will be applied to analyze the contribution of each feature to individual predictions in the top-performing model to provide global explanations by identifying overall churn drivers and local explanations by explaining individual churn predictions. Partial Dependence Plots (PDP) and Individual Conditional Expectation (ICE) plots will be used to visualize key relationships between top features and churn outcomes. As a result, model transparency will be improved, and the experimental results can be interpreted in a more in-depth manner.

Experimental Environment: All experiments will be conducted using Python-based data science tools. Libraries such as NumPy, pandas, and scikit-learn will be used for data processing, model training, and evaluation. The entire project will be carried out in the Google Colab environment, which supports collaborative development and helps ensure transparency and accessibility for all project contributors.

7 Proposed Evaluation Metrics

This project is developed with the aim of predicting customer churn risk and supporting businesses in reducing this risk through appropriate strategies based on the predictions. Machine learning models will be used for this purpose, and evaluating the performance of these models during the prediction process is of great importance. As previously mentioned, since the target variable in this project consists of two values, 0 and 1, the problem is formulated as a binary classification task. The success of the predictions will not be evaluated solely based on overall accuracy, but also by using evaluation metrics that take different types of prediction errors into account.

Accuracy: During model evaluation, the accuracy metric will first be reported. Accuracy, as its name suggests, measures the proportion of correct predictions made by the models. However, as discussed earlier in this project, the class distribution of the target variable is not completely balanced. The target values are distributed with proportions of approximately 54% and 46%, which indicates a slight class imbalance. Due to this imbalance, a model may still achieve a relatively high accuracy score while failing to perform well in identifying important churn-related patterns. Therefore, there is a need to consider evaluation metrics that focus not only on correct predictions but also on the types of errors made by the models.

Confusion Matrix: In order to analyze prediction errors in more detail, the use of a confusion matrix is planned. By examining the values of True Positive, True Negative, False Positive, and False Negative, the confusion matrix enables an analysis of which types of errors are made more frequently by the model. In the context of this project, a high false negative rate, where customers with a high likelihood of leaving are incorrectly predicted as staying, is considered the most critical type of error. Therefore, analyzing this error type is essential for the success of the project.

Precision, Recall and F1-Score: To further evaluate model performance, precision and recall metrics will be used to analyze the trade-off between false alarms, where customers who are unlikely to leave are predicted as churners, and missed churn cases, where customers who are likely to leave are incorrectly predicted as non-churners. In addition, the balance between these two metrics will be summarized using the F1-score, which represents the harmonic mean of precision and recall. In this analysis, higher F1-score values are expected to indicate better model performance, while lower values suggest a weaker balance between precision and recall. Accordingly, the F1-score will be used to assess how well the model balances these two metrics.

ROC-AUC: Since the churn dataset does not exhibit a severe class imbalance, the probability of churn will also be evaluated using the ROC-AUC score. This metric provides a single score that reflects the model's ability to distinguish between churn and non-churn classes across different decision thresholds. ROC-AUC will be particularly useful during the comparison of different models. Accordingly, achieving a higher ROC-AUC score will indicate that the model has a stronger ability to separate churn and non-churn classes, whereas values approaching 0.5 will reflect weaker discriminative performance.

All of these evaluation metrics will be reported on the validation dataset for the purpose of model selection and comparison. In this way, the suitability of the developed models for churn risk prediction will be assessed using multiple evaluation criteria. Finally, the selected models will be evaluated on the test dataset, and the final performance metrics will be reported. In this way, it can be tested whether the models are able to produce unbiased, fair, and reliable results on unseen data during the model development process.

8 Project Execution Timeline

The following timeline summarizes the main project tasks, milestones, and expected completion schedule.

Week	**Tasks**	**Target Completion Date**
Week 1	Dataset understanding, data cleaning, EDA	10 Feb 2026
Week 2	Feature engineering and baseline model	17 Feb 2026
Week 3	Advanced models and hyperparameter tuning	24 Feb 2026
Week 4	Model evaluation, interpretation	3 Mar 2026
Week 5	Report writing	10 Mar 2026

9 References

References

[1] Sarker I. H. (2021). *Machine Learning: Algorithms, Real-World Applications and Research Directions*. SN computer science, 2(3), 160.

[2] I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam, & S. W. Kim (2019). *A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn*. IEEE Access, vol. 7.

[3] Pawan Trivedi (2022). *Customer Churn*. Kaggle vol. 1

[4] Shaaban, E., Helmy, Y., Khedr, A., & Nasr, M. (2012). *A proposed churn prediction model.* International Journal of Engineering Research and Applications (IJERA), 2(4), 693–697.

[5] Manzoor, A., Qureshi, M. A., Kidney, E., & Longo, L. (2024). *A review on machine learning methods for customer churn prediction and recommendations for business practitioners. IEEE Access*, 12, 70434–70450. `https://doi.org/10.1109/ACCESS.2024.3402092`