Machine Learning Using Trees Analysis Report

Contents

Project and Dataset Selection

The objective of this project is to develop and compare four tree-based machine learning models: Decision Tree, Random Forest, AdaBoost, and XGBoost to predict whether a Spotify user will churn.

In this context, user churn refers to a customer discontinuing use of Spotify's services or canceling their subscription. Accurately identifying patterns that distinguish churned users from active ones allows Spotify to design targeted retention strategies, improve personalized recommendations, and reduce overall cancellation rates.

The dataset used for this project is the Kaggle Spotify Churn Analysis Dataset, which contains approximately 8,000 observations and 12 behavioral and demographic features describing user activity and subscription details. Examples of these features include metrics such as listening time, skip rate, age, and songs played per day.

This task is formulated as a binary classification problem, where:
is_churned = 0 → Active user
is_churned = 1 → Churned user

Beyond model prediction, this analysis also seeks to identify which user behavior and demographic attributes most strongly influence churn risk, providing actionable insights into customer engagement and retention. The ultimate goal is to determine which tree-based algorithm delivers the highest predictive performance and interpret the key factors that contribute to user cancellations.

Regression Model Building

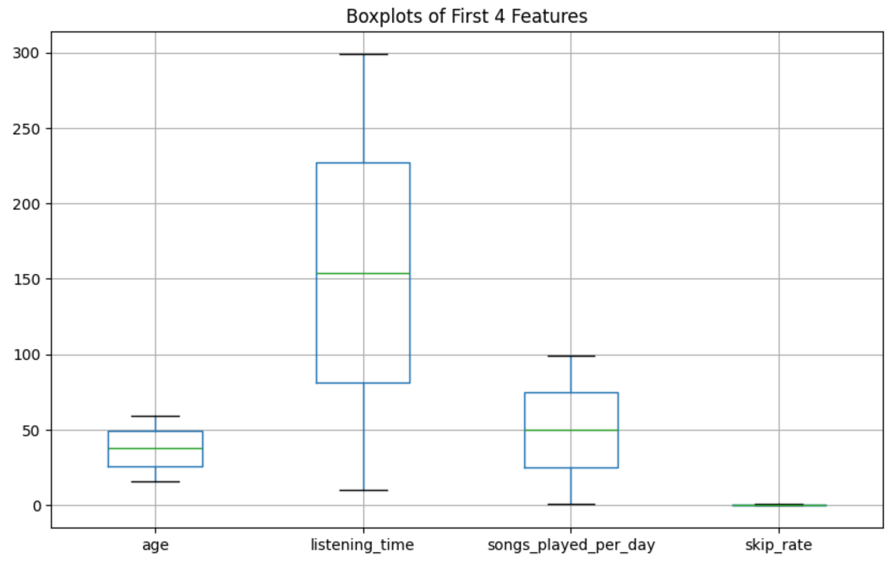
Data Pre-Processing

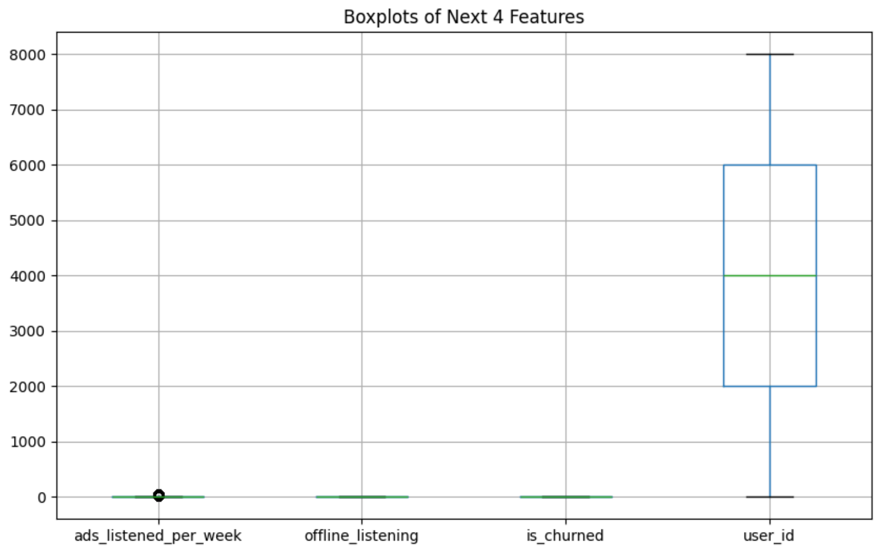
The dataset was loaded into a Pandas DataFrame. A new DataFrame *Spotify* was created.

To ensure data consistency I first checked for null values using the .isnull().sum() function. It was confirmed that all 12 columns contain 8,000 non-null entries, indicating there are no missing values. The .describe() summary confirmed that numeric attributes fall within realistic ranges and do not display significant outliers.

User ages range from 16 to 59 years, listening time varies from 10 to 299 minutes per day, and skip rates remain within the expected 0.0–0.6 interval. Categorical attributes such as gender, country, subscription_type, and device_type contain a small, consistent set of valid categories without irregular entries. All user IDs are unique, confirming no duplicated data. Overall, the dataset is clean, consistent, and ready for feature transformation and modeling.

Then I proceeded to visualize the distribution and skewness of the numerical attributes by using boxplots and histograms. The boxplots provided a clear view of the spread of each variable and potential outliers.

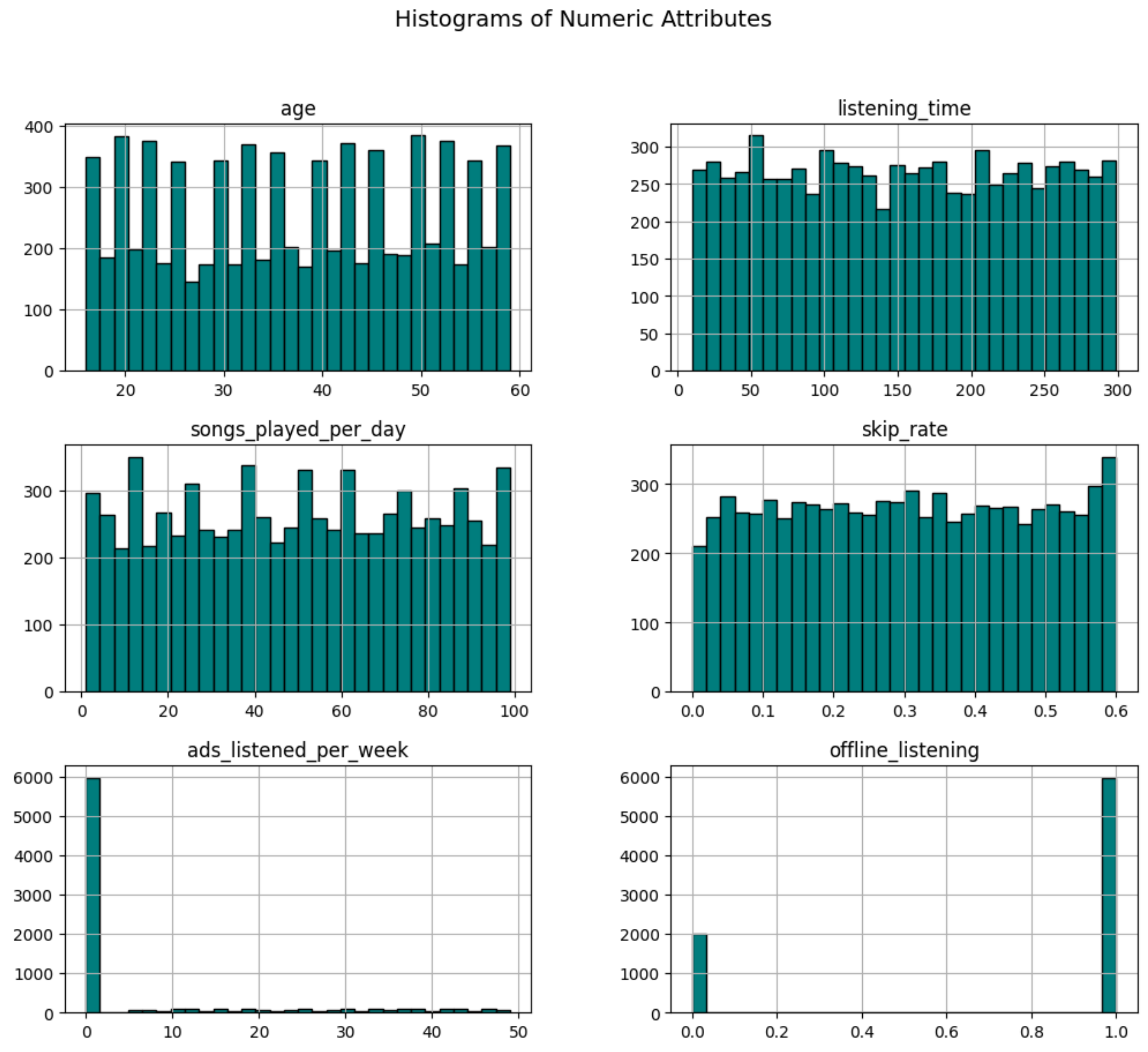




The boxplots revealed that the data does not contain any significant outliers, as there were no points extending beyond the whiskers for most features. This indicates that all attributes fall within reasonable ranges, confirming that the dataset is clean and free of extreme anomalies.

Variables such as listening_time, songs_played_per_day, and ads_listened_per_week showed wider boxes, reflecting greater variability among users, while skip_rate and offline_listening displayed narrower spreads.

Next, I plotted histograms to further explore the shape and skewness of the data.
The histograms showed that most numerical attributes are not normally distributed.
Features such as age, listening_time, and songs_played_per_day appear roughly uniform, while ads_listened_per_week is heavily right-skewed, with a large concentration of users who hear few or no ads and a small subset of users with very high ad counts. The offline_listening variable is binary, leading to two tall spikes at 0 and 1, which is expected.



Examining Attributes and Target Variables

The dataset contains a mix of numerical and categorical variables that describe Spotify users' demographics, engagement behaviors, and subscription details. Each observation represents a unique user, and the attributes capture information such as listening patterns, device usage, and account type.

The main attributes include:

- gender – User gender (Male, Female, or Other)
- age – User's age in years
- country – Country of residence (e.g., US, CA, DE, AU, etc.)
- subscription_type – Type of Spotify plan (Free, Premium, Family, or Student)
- listening_time – Average daily listening time in minutes
- songs_played_per_day – Number of songs played daily
- skip_rate – Proportion of songs skipped (0–1 scale)
- device_type – Primary device used (Mobile, Desktop, or Web)
- ads_listened_per_week – Number of advertisements heard per week
- offline_listening – Binary indicator (1 = uses offline mode, 0 = does not)
- is_churned – Target variable; 1 = churned (inactive user), 0 = active user

Categorical values (gender, country, subscription_type, device_type) were converted into numerical format to ensure compatibility with tree-based models. Binary variables such as offline_listening and the target variable is_churned were already numeric and required no transformation.

Standardization and Normalization of the attributes

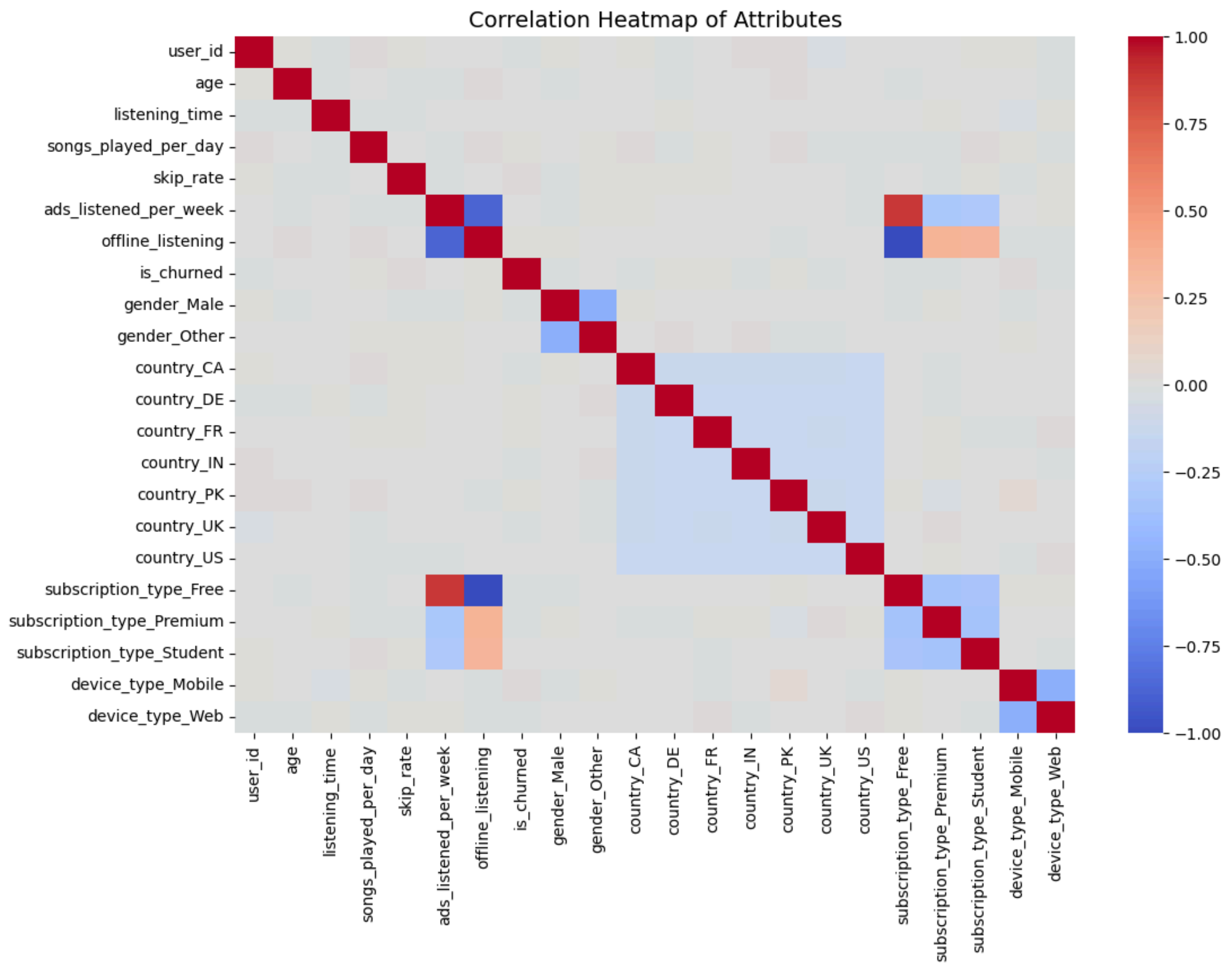
Now, to prepare the dataset for modeling, all numerical attributes were standardized and normalized to ensure that they are on comparable scales. Although tree-based algorithms such as Decision Tree, Random Forest, AdaBoost, and XGBoost are not sensitive to feature scaling, these transformations were performed for consistency and to facilitate comparison across models.

Standardization was applied first, transforming each attribute to have a mean of 0 and a standard deviation of 1. This was followed by normalization, which rescaled the features into the [0,1] range. Both transformations were performed using StandardScaler and MinMaxScaler from the Scikit-learn library. The standardized and normalized datasets were stored separately to allow flexibility in model evaluation.

Correlation Analysis and Feature Importance

To understand how the different variables relate to each other and to the target variable is_churned, Pearson correlation coefficients were calculated for all attributes. The resulting correlations were mostly weak, suggesting that each feature contributes independently to predicting user churn.

The three features most positively correlated with churn were skip_rate (0.016), device_type_Mobile (0.016), and country_PK (0.014). This indicates that users who skip songs frequently, primarily use mobile devices, or are from specific regions are slightly more likely to stop using Spotify. Conversely, listening_time (-0.0076) and songs_played_per_day (0.0093) showed near-zero but directionally intuitive correlations, implying that users who listen longer and play more songs tend to stay active.



Based on this analysis, the most relevant attributes for predicting user churn are skip_rate, listening_time, and songs_played_per_day, as they directly represent user engagement behavior. These variables are likely to have the most predictive power when training classification models, while demographic variables such as age, gender, and country appear less influential.

Model Construction

Four tree-based classification models were implemented in Python: Decision Tree, Random Forest, AdaBoost, and XGBoost. All models were developed using the Scikit-learn and XGBoost libraries.

To ensure fair comparison and optimal performance, each model was tuned using GridSearchCV with 5-fold cross-validation. This approach automatically trained and evaluated every combination of specified hyperparameters, selecting the configuration that produced the highest mean cross-validated accuracy.

Decision Tree

For the Decision Tree, the model was initialized with a fixed random state for reproducibility, and its parameter grid explored variations in:

- criterion: to compare the Gini and entropy impurity measures
- max_depth: to limit how deep each tree could grow
- min_samples_split: the minimum number of samples required to split a node
- min_samples_leaf: the minimum number of samples allowed in a leaf node

Random Forest

The Random Forest model extended this concept by building an ensemble of multiple Decision Trees. Its grid included:

- n_estimators: number of trees in the forest
- max_depth, min_samples_split, and min_samples_leaf: to control tree structure
- criterion: to test both Gini and entropy criteria. The grid search evaluated all parameter combinations with 5-fold CV, using parallel computation (n_jobs = -1) to reduce runtime

AdaBoost

For AdaBoost, the model used a base ensemble of shallow trees trained sequentially. The hyperparameters tuned were:

- n_estimators: number of weak learners
- learning_rate: the contribution of each weak learner to the final prediction

XGBoost

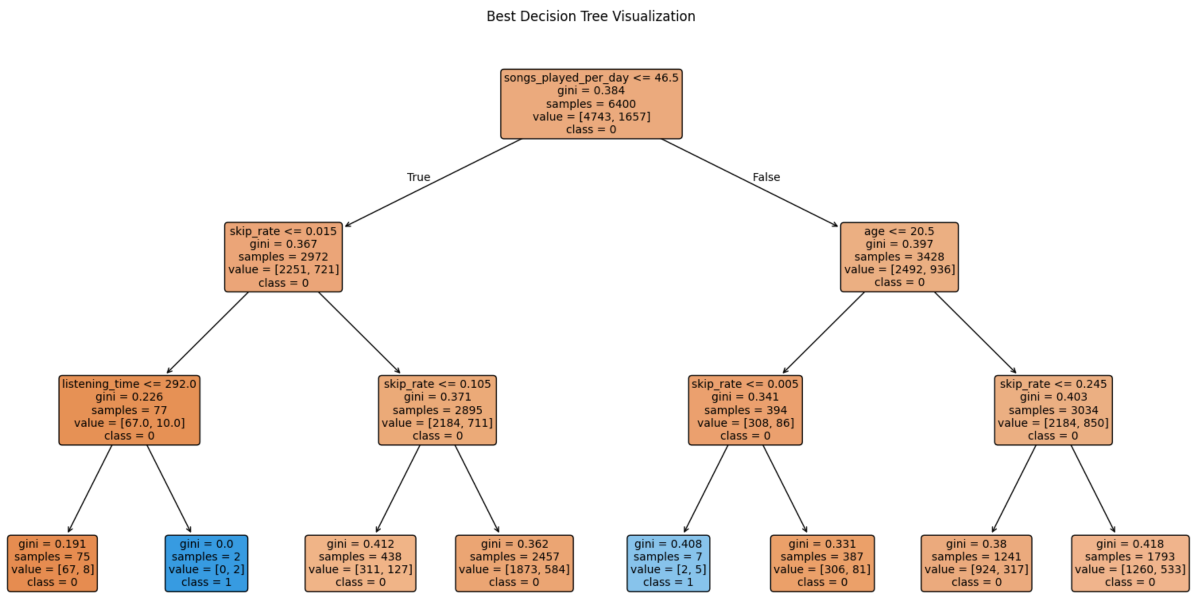
Finally, the XGBoost classifier was configured with:

- n_estimators
- max_depth
- learning_rate
- subsample: controlling the fraction of data used for each boosting round

Across all models, GridSearchCV evaluated each parameter combination over five folds of the training data, computing the average accuracy for each setting. The configuration that achieved the highest mean cross-validated accuracy was stored as best_params_, representing the tuned version of the model to be compared in later evaluation.

Model Visualization

A visualization of the best Decision Tree model was generated using Scikit-learn’s plot_tree() function. The diagram displays the hierarchical structure of the model, showing how the dataset is split at each node according to feature thresholds. Each node contains information about the feature used for the split, the Gini impurity, the number of samples reaching that node, the class distribution, and the predicted class.



From the tree, it can be observed that the feature songs_played_per_day serves as the root split, indicating its high importance in predicting the target variable. Subsequent splits on features such as skip_rate, age, and listening_time further refine the decision boundaries. The color shading illustrates the purity of each node, darker nodes represent purer classifications. This visualization provides interpretability by clearly showing how the model arrives at its predictions and which attributes most strongly influence classification outcomes. A similar visualization was also generated for one of the Random Forest's individual trees to illustrate the structure of its component learners.

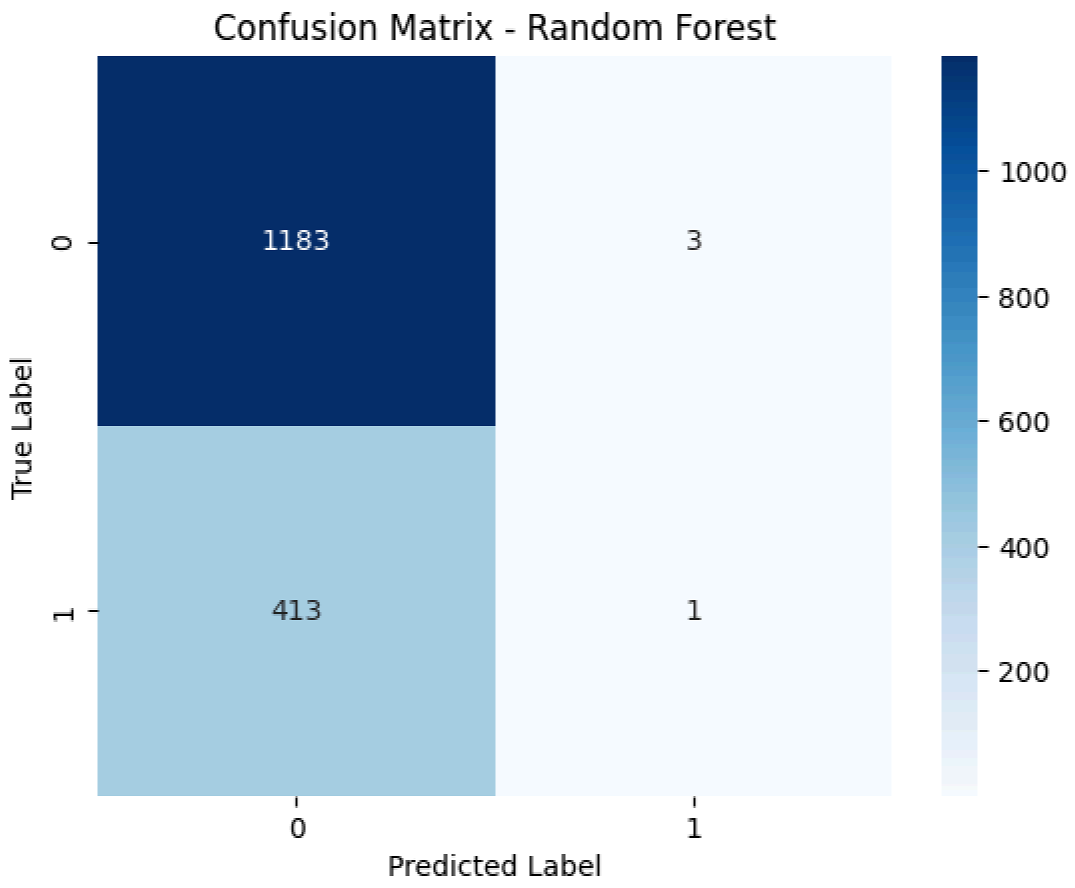
Analysis

To evaluate model performance beyond simple accuracy, a detailed analysis was performed using additional metrics and visual diagnostics. These included the confusion matrix, precision, recall, F1-score, and graphical evaluations through the ROC and Precision–Recall curves.

Confusion Matrix and Classification Matrix

The confusion matrix illustrates the distribution of predictions across the true classes, highlighting both correctly classified and misclassified samples. The best-performing Random Forest model was used for this detailed evaluation since it achieved the highest validation accuracy.

From the confusion matrix, it was observed that the model predicted the majority class with high precision and recall, but struggled to correctly classify instances of the minority class. The classification report confirmed this imbalance:



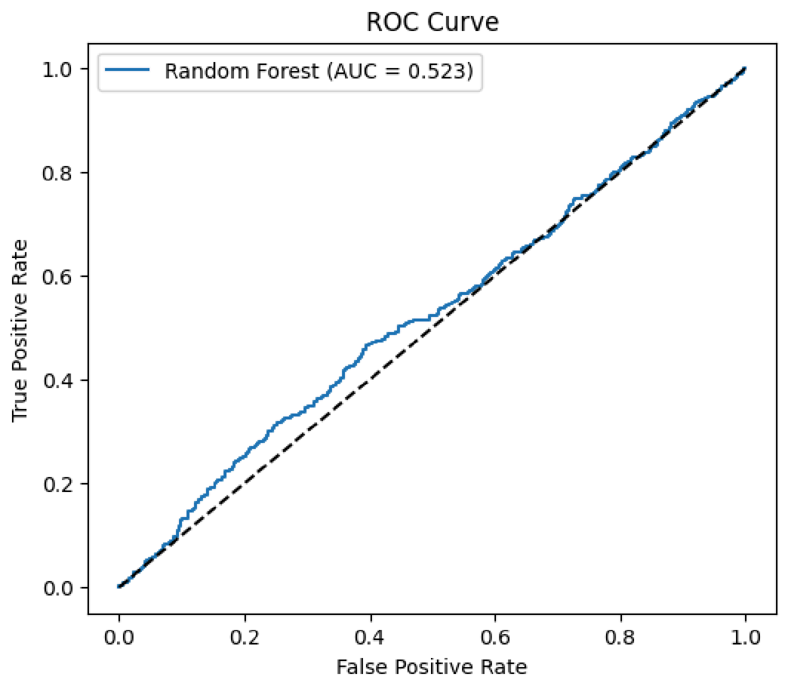
	precision	recall	f1-score	support
0	0.74	1.00	0.85	1186
1	0.25	0.00	0.00	414
accuracy			0.74	1600
macro avg	0.50	0.50	0.43	1600
weighted avg	0.61	0.74	0.63	1600

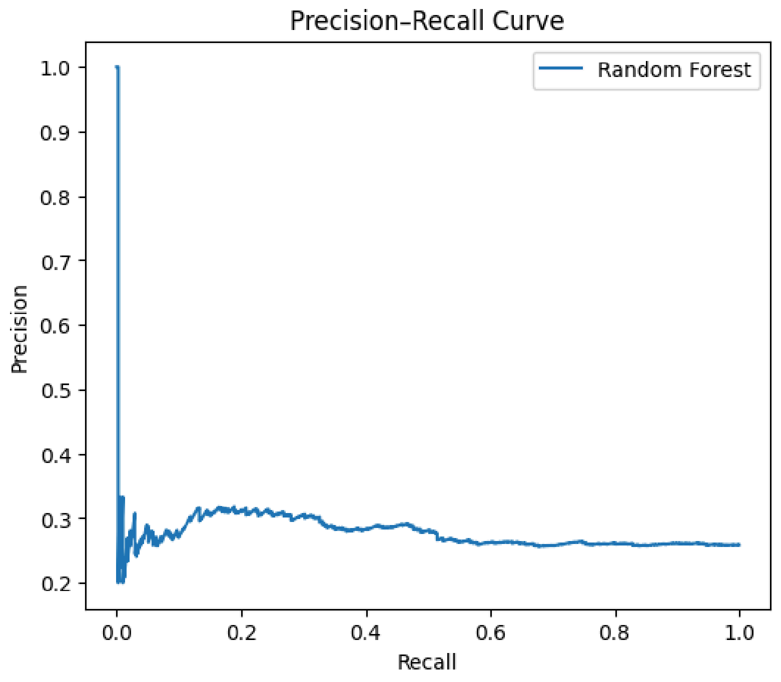
The Random Forest achieved high precision (0.74) and perfect recall (1.00) for class 0, but very low precision (0.25) and near-zero recall for class 1. This indicates that while the model performs well overall in terms of accuracy (0.74), it fails to generalize effectively for minority cases, resulting in a low macro-averaged F1-score. Such behavior suggests a bias toward the dominant class, a common issue in real-world datasets with uneven class distributions.

ROC Curve and Precision

The ROC curve visualizes the trade-off between the True Positive Rate and False Positive Rate. The Area Under the Curve was approximately 0.523, indicating that the Random Forest performs only marginally better than random guessing when distinguishing between positive and negative samples.

Similarly, the Precision–Recall curve demonstrates that precision remains low across most recall levels, confirming the limited predictive power for the minority class. These findings reinforce that accuracy alone does not fully represent model quality, particularly in the presence of class imbalance.





Comparison of All Models

All four tree-based models were tuned using GridSearchCV and compared based on cross-validated accuracy and extended metrics.

The Decision Tree achieved an average accuracy of 0.7411, showing solid baseline performance but with susceptibility to overfitting.

The Random Forest, with an accuracy of 0.7416, offered slightly better generalization through ensemble averaging and was the most consistent across validation folds.

AdaBoost and XGBoost achieved comparable accuracies (approximately 0.7411) but did not outperform the Random Forest, suggesting that further boosting did not yield measurable improvement given the feature set and data characteristics.

Despite the small numerical differences, the Random Forest emerged as the most balanced and reliable model overall. Its ensemble structure effectively reduced variance compared to a single tree while maintaining interpretability and robustness. However, all models showed limitations in minority-class detection, emphasizing that accuracy alone can be misleading in imbalanced settings. Future work could address this through resampling methods or the use of class-weighted loss functions.

Conclusion

This project implemented and compared four tree-based machine learning models for predicting Spotify user churn. After extensive hyperparameter tuning and performance evaluation using cross-validation, confusion matrices, and ROC/PR curves, the Random Forest model emerged as the most consistent and generalizable classifier. Although overall accuracy was moderate (≈74%), the analysis revealed challenges with minority-class prediction, highlighting opportunities for future improvement through data balancing, feature selection, or advanced ensemble techniques.