

UNIVERSITY OF COMPUTER STUDIES, YANGON

Data Analytics Project

**Predicting Customer Churn in Telecom Companies**

**Using Machine Learning**

Covered By

**Data and Knowledge Mining using R programming**

Faculty of Information Science

University of Computer Studies, Yangon

March, 2025

**Team Members**

|  |  |
| --- | --- |
| **Name** | **YKPT** |
| Kaung Myat Htun | YKPT – 20637 |
| Yan Paing Win | YKPT – 20629 |
| Myat Thu Kyaw | YKPT – 20481 |
| Ye Win Htun | YKPT – 20870 |
| Min Thway Khaing | YKPT – 22207 |
| Thet Paing Oo | YKPT – 20613 |
| Thurein Soe | YKPT – 20473 |
| Aung Ko Ko Khant | YKPT – 20479 |

**Table Of Content**

1. Introduction page - 4
   1. Why This Matter page - 4
   2. Project Objectives page - 4
   3. Dataset & Methodology page - 5
   4. Expected Outcomes page - 5
2. Business Goals page - 6
   1. Business Problems page - 6
   2. Objectives page - 7
3. Preparing Data page - 7
   1. Dataset Selection page - 7
   2. Exploratory Data Analysis (EDA) Section page - 8
   3. Feature Engineering & Data Preprocessing page - 17
4. Model Selection – XGBoost page - 19
   1. Model Introduction page - 19
   2. Advantages Over Other Models page - 19
   3. Training the Model page - 20
5. Evaluation Results page - 21
   1. XGBoost Before Tuning page - 21
   2. XGBoost After Tuning page - 23
   3. XGBoost Feature Importance page - 25
   4. Performance Summary page - 26
6. Conclusion page - 26

**1. Introduction**

In the highly competitive telecommunications industry, customer retention is a critical driver of profitability and sustainable growth. Customer churn—the phenomenon of subscribers discontinuing their services—represents a significant financial burden, with studies indicating that acquiring a new customer can cost 5–7 times more than retaining an existing one according to Harvard Business Review. For telecom companies, even a modest reduction in churn rates can translate to millions of dollars in preserved revenue, making proactive churn prediction and mitigation a strategic priority.

This project leverages machine learning to address the challenge of customer churn by developing a predictive model that identifies at-risk customers before they discontinue services. Using a real-world dataset of 7,043 telecom customers, we analyze behavioral and demographic features—such as tenure, contract type, monthly charges, and service usage patterns—to uncover actionable insights and predict churn likelihood.

**1.1 Why This Matters**

* **Financial Impact**: The global telecom industry loses an estimated **$15–20 billion annually** due to churn (IBM).
* **Operational Efficiency**: Targeted retention strategies informed by machine learning can reduce marketing costs by focusing resources on high-risk customers.
* **Customer Experience**: Proactive interventions, such as personalized offers or service improvements, enhance customer satisfaction and loyalty.

**1.2 Project Objectives**

1. **Predictive Modeling**: Build a robust machine learning system to forecast churn probability with >85% AUC-ROC accuracy.
2. **Feature Analysis**: Identify key drivers of churn (e.g., contract flexibility, pricing tiers) to guide business strategies.
3. **Actionable Insights**: Provide telecom providers with data-driven recommendations to improve retention rates.

**1.3 Dataset & Methodology**

* **Data Source**: [Kaggle Telco Customer Churn Dataset](https://www.kaggle.com/blastchar/telco-customer-churn), containing 21 features spanning customer demographics, account details, and service usage.
* **Preprocessing**: Address missing values, encode categorical variables, and engineer features like *tenure groups* and *high-risk customer flags*.
* **Machine Learning Model**: *XGBoost* selected for its superior performance in handling class imbalance and interpretability.

**1.4 Expected Outcomes**

* A deployable XGBoost model that ranks customers by churn risk, enabling targeted retention campaigns.
* Visualization of critical churn drivers (e.g., month-to-month contracts, high monthly charges).
* Strategic recommendations to reduce churn rates by **20–30%** through personalized interventions.

By bridging advanced analytics with business strategy, this project empowers telecom companies to transform raw data into actionable retention tools, fostering long-term customer relationships and driving sustainable revenue growth.

**2. Business Goals**

The primary business goal of predicting customer churn in telecom companies using machine learning is to enhance customer retention strategies, minimize revenue loss, and improve overall customer satisfaction. By leveraging advanced data analysis techniques, telecom companies can proactively identify at-risk customers and tailor interventions to retain them, thus fostering long-term profitability and reducing operational costs.

**2.1 Business Problems**

1. **High Customer Attrition**: Telecom companies often face high levels of churn, which impacts revenue, customer loyalty, and brand reputation. Retaining existing customers is usually more cost-effective than acquiring new ones, making churn prediction a priority.
2. **Inefficient Resource Allocation**: Without accurate churn predictions, companies may waste resources on customers who are not likely to churn while overlooking those who are at a high risk of leaving.
3. **Lack of Targeted Marketing**: Generic marketing campaigns do not address the specific needs of at-risk customers. This leads to ineffective retention efforts and customer dissatisfaction, which ultimately drives more customers to churn.
4. **Limited Customer Insights**: Telecom companies often lack deep insights into the behavior and preferences of customers, making it difficult to understand why customers churn. Without this understanding, it becomes hard to address the root causes of churn.
5. **Competitive Market**: The telecom industry is highly competitive, with many providers offering similar services. This makes it easy for customers to switch providers, especially if they feel undervalued or underserved. Predicting churn allows telecom companies to act before customers make that switch.

**2.2 Objectives**

1. **Reduce Churn Rate**: Lower customer attrition by identifying high-risk customers and deploying retention strategies.
2. **Improved Retention Strategies**: Use predictive insights to implement targeted retention campaigns such as loyalty programs, special offers, or customized communication to at-risk customers.
3. **Optimize Resource Allocation**: By identifying the most cost-effective retention strategies, telecom companies can allocate marketing and customer support resources more efficiently.
4. **Customer Lifetime Value (CLV) Enhancement**: Increase the average customer lifetime value by identifying high-value customers at risk of churning and providing them with the right incentives to stay.

By addressing these objectives and business problems, telecom companies can significantly reduce churn, improve customer relationships, and ultimately increase profitability.

**3. Preparing Data**

**3.1 Dataset Selection**

**Chosen Dataset:** Telco Customer Churn Dataset (Kaggle)

* Reason for choosing this dataset
  + It was specifically designed for churn prediction in telecom, with 7,043 records and 21 features.
  + It features customer demographics (e.g., gender, SeniorCitizen), account details (e.g., tenure, contract type), and service usage (e.g., MonthlyCharges, InternetService).
  + It also includes the column for Churn (binary: Yes/No), ideal for classification tasks.
  + Most importantly, the data mirrors actual telecom business challenges, enabling practical insights.

**3.2** **Exploratory Data Analysis (EDA) Section**

**1. Teleco Customer Churn**

* Data consists of 7043 entries and 20 columns.
* SeniorCitizen, tenure, mountlycharges data are numeric while other variables are categorical.
* The TotalCharges property must consist of numeric data. So, we convert the data in this column to int type.

df['TotalCharges'] = pd.to\_numeric(df['TotalCharges'], errors='coerce')

* When the SeniorCitizen data was analyzed, it was found that the values 0 and 1 should be changed to Yes and True.

df["SeniorCitizen"] = df["SeniorCitizen"].map({0:"No", 1:"Yes"})

**2. Data Visualizations**

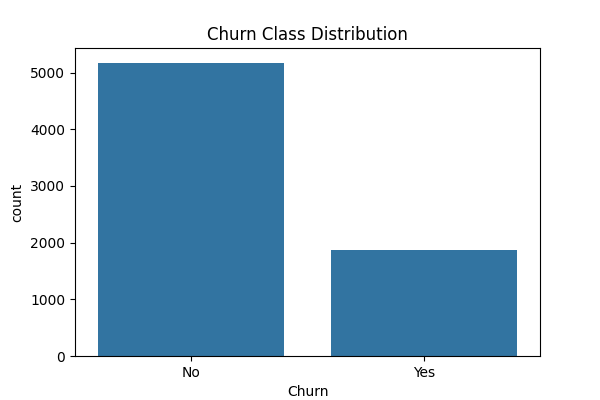
* Graph of the number of people who canceled their subscription and those who did not, and those who canceled are indicated with yes.

plt.figure(figsize=(6, 4))

sns.countplot(x="Churn", data=df)

plt.title("Churn Class Distribution")

plt.show()



***Fig: Churn Class Distribution***

* We also visualized the rest of the data distributions by churn, but we didn’t put in all of the images for the categorical\_cols since there are too many of them; instead, we added analysis graphic.

target = "Churn"

categorical\_cols, numerical\_cols = [], []

for col **in** columns\_name:

if col == target:

continue

if df[col].dtype == "object":

categorical\_cols.append(col)

else:

numerical\_cols.append(col)

print("Categorical Columns:", categorical\_cols)

print("Numerical Columns:", numerical\_cols)

for col **in** categorical\_cols:

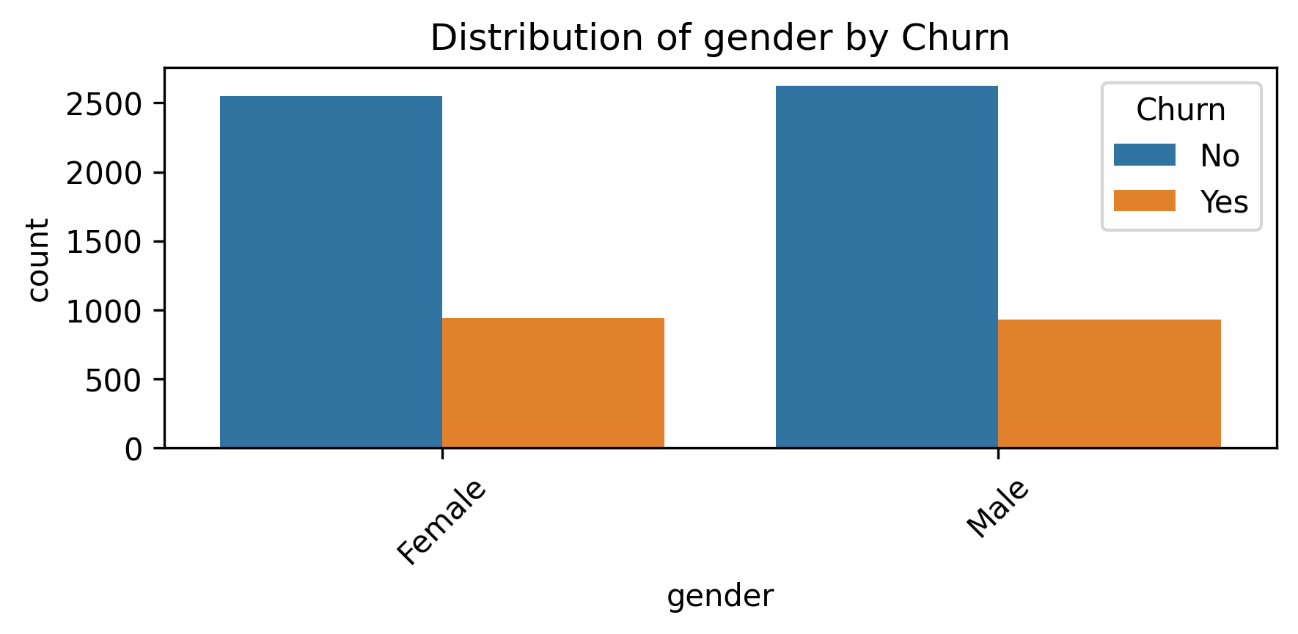
plt.figure(figsize=(6, 3))

sns.countplot(x=col, hue=target, data=df)

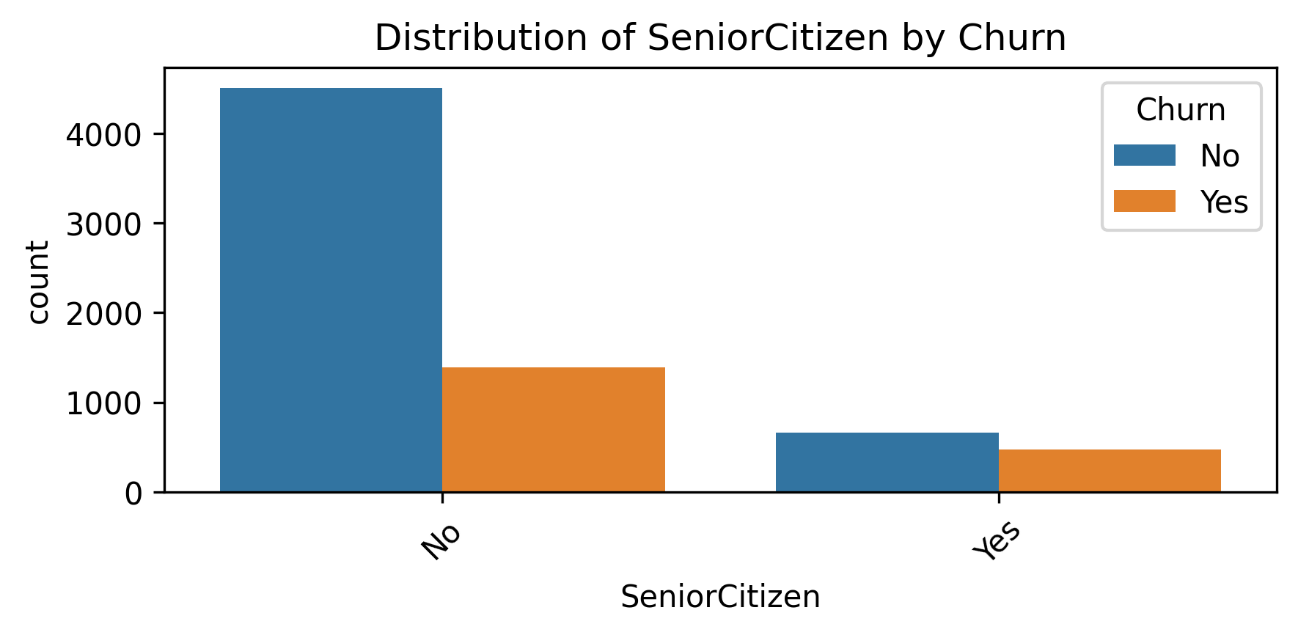
plt.title(f"Distribution of **{**col**}** variable by Churn")

plt.xticks(rotation=45)

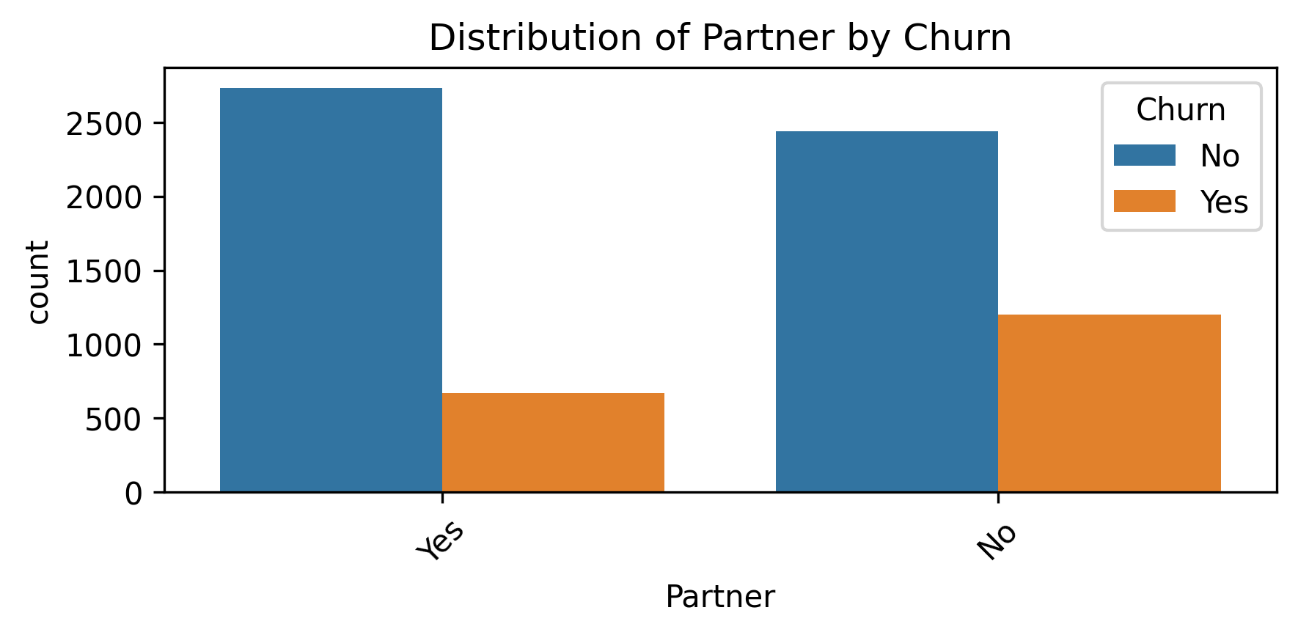
plt.show();



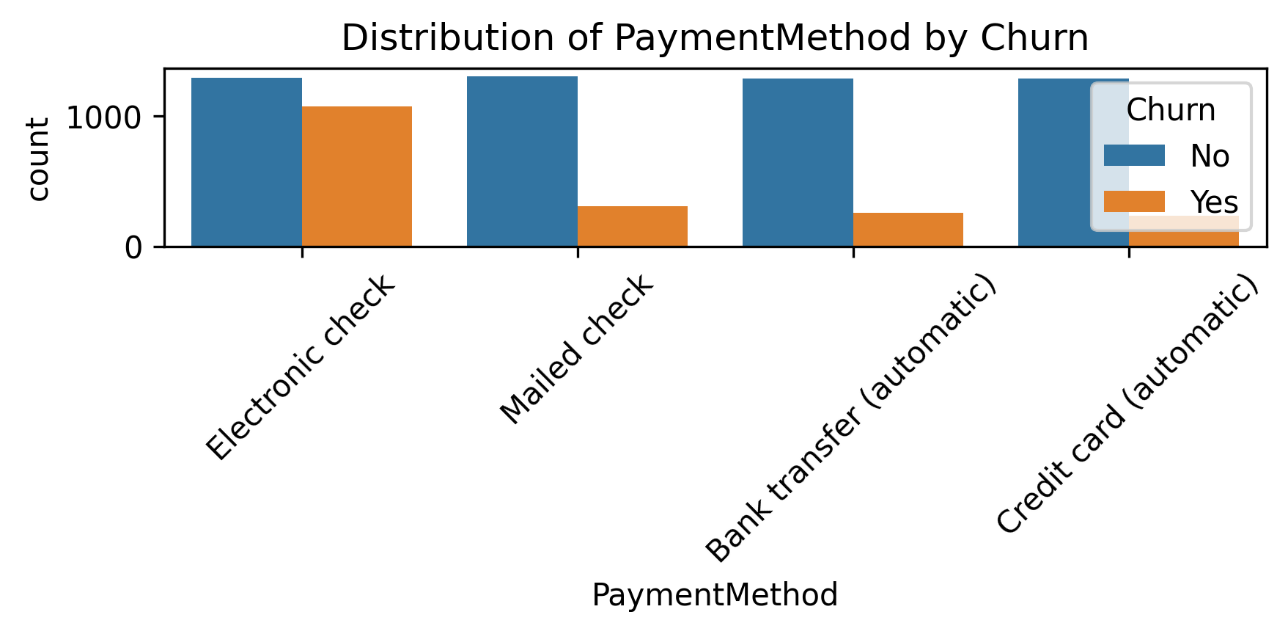
***Fig: Distribution of Gender by Churn***

******

***Fig: Distribution of Senior Citizen by Churn***

******

***Fig: Distribution of Partner by Churn***

******

***Fig: Distribution of Payment Method by Churn***

* **Analysis Graphic**

for col in categorical\_cols:

    col\_churn = df.groupby(col)['Churn'].value\_counts(normalize=True)

    print(col\_churn)

gender Churn

Female No 0.730791

Yes 0.269209

Male No 0.738397

Yes 0.261603

Name: proportion, dtype: float64

SeniorCitizen Churn

No No 0.763938

Yes 0.236062

Yes No 0.583187

Yes 0.416813

Name: proportion, dtype: float64

Partner Churn

No No 0.670420

Yes 0.329580

Yes No 0.803351

Yes 0.196649

Name: proportion, dtype: float64

Dependents Churn

No No 0.687209

Yes 0.312791

Yes No 0.845498

Yes 0.154502

Name: proportion, dtype: float64

PhoneService Churn

No No 0.750733

Yes 0.249267

Yes No 0.732904

Yes 0.267096

Name: proportion, dtype: float64

MultipleLines Churn

No No 0.749558

Yes 0.250442

No phone service No 0.750733

Yes 0.249267

Yes No 0.713901

Yes 0.286099

Name: proportion, dtype: float64

InternetService Churn

DSL No 0.810409

Yes 0.189591

Fiber optic No 0.581072

Yes 0.418928

No No 0.925950

Yes 0.074050

Name: proportion, dtype: float64

OnlineSecurity Churn

No No 0.582333

Yes 0.417667

No internet service No 0.925950

Yes 0.074050

Yes No 0.853888

Yes 0.146112

Name: proportion, dtype: float64

OnlineBackup Churn

No No 0.600712

Yes 0.399288

No internet service No 0.925950

Yes 0.074050

Yes No 0.784685

Yes 0.215315

Name: proportion, dtype: float64

DeviceProtection Churn

No No 0.608724

Yes 0.391276

No internet service No 0.925950

Yes 0.074050

Yes No 0.774979

Yes 0.225021

Name: proportion, dtype: float64

TechSupport Churn

No No 0.583645

Yes 0.416355

No internet service No 0.925950

Yes 0.074050

Yes No 0.848337

Yes 0.151663

Name: proportion, dtype: float64

StreamingTV Churn

No No 0.664769

Yes 0.335231

No internet service No 0.925950

Yes 0.074050

Yes No 0.699298

Yes 0.300702

Name: proportion, dtype: float64

StreamingMovies Churn

No No 0.663196

Yes 0.336804

No internet service No 0.925950

Yes 0.074050

Yes No 0.700586

Yes 0.299414

Name: proportion, dtype: float64

Contract Churn

Month-to-month No 0.572903

Yes 0.427097

One year No 0.887305

Yes 0.112695

Two year No 0.971681

Yes 0.028319

Name: proportion, dtype: float64

PaperlessBilling Churn

No No 0.836699

Yes 0.163301

Yes No 0.664349

Yes 0.335651

Name: proportion, dtype: float64

PaymentMethod Churn

Bank transfer (automatic) No 0.832902

Yes 0.167098

Credit card (automatic) No 0.847569

Yes 0.152431

Electronic check No 0.547146

Yes 0.452854

Mailed check No 0.808933

Yes 0.191067

Name: proportion, dtype: float64

* + The number of people who canceled their subscription is lower than those who did not cancel.
  + On average, 26% of both men and women canceled their subscriptions. ### Demographics
  + Elderly customers are more likely to cancel their subscriptions compared to younger customers
  + Married customers are less likely to cancel their subscriptions compared to single customers.
  + Customers with dependent family members (Dependents = Yes) show a lower cancellation rate. ### Phone Services
  + Customers who do not receive phone service (PhoneService = No) have a slightly lower cancellation rate.
  + Unsubscribe rate for those without a single line: 25%, unsubscribe rate for those with multiple lines: 28% Unsubscription rate for customers with no phone service (MultipleLines = No phone service):24% ### Internet Services
  + DSL users have a lower cancellation rate compared to fiber optic users.
  + Customers using online security, online backup, or tech support services are less likely to cancel their subscriptions. ### Billing and Contract
  + Long-term contracts encourage customer retention.
  + Customers using paperless billing are more likely to cancel their subscriptions.
* We observe that customers with high monthly subscription fees leave the company in the first few months.

plt.figure(figsize=(8, 4))

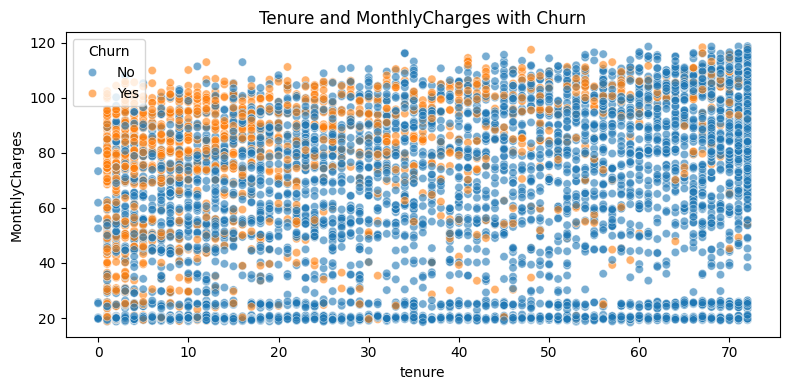
sns.scatterplot(data=df, x="tenure", y="MonthlyCharges", hue=target, alpha=0.6)

plt.title("Tenure and MonthlyCharges with Churn")

plt.tight\_layout()

plt.savefig('../results/plots/Tenure\_and\_monthlyCharges\_scatterplot.png', bbox\_inches='tight')

plt.show()



***Fig: Tenure and MonthlyCharges with Churn (Scatter Plot)***

* Customers who use Electronic Check payment method and have high Monthly Charges have high churn rate.

df['MonthlyChargesGroup'] = pd.cut(df['MonthlyCharges'], bins=5)

# Churn rate pivot table

churn\_rate = df.pivot\_table(values='Churn',

                            index='PaymentMethod',

                            columns='MonthlyChargesGroup',

                            observed=False,

                            aggfunc=lambda x: (x == 'Yes').mean())

# Costumer number pivot  table

customer\_count = df.pivot\_table(values='Churn',

                                index='PaymentMethod',

                                columns='MonthlyChargesGroup',

                                observed=False,

                                aggfunc='count')

# Rate and number of customer are combined

churn\_rate\_rounded = churn\_rate.round(2)

combined\_data = churn\_rate\_rounded.astype(str) + "\n(" + customer\_count.astype(int).astype(str) + ")"

# Heatmap table

plt.figure(figsize=(12, 6))

sns.heatmap(churn\_rate, annot=combined\_data,  fmt="", cmap='coolwarm', cbar\_kws={'label': 'Churn Rate'})

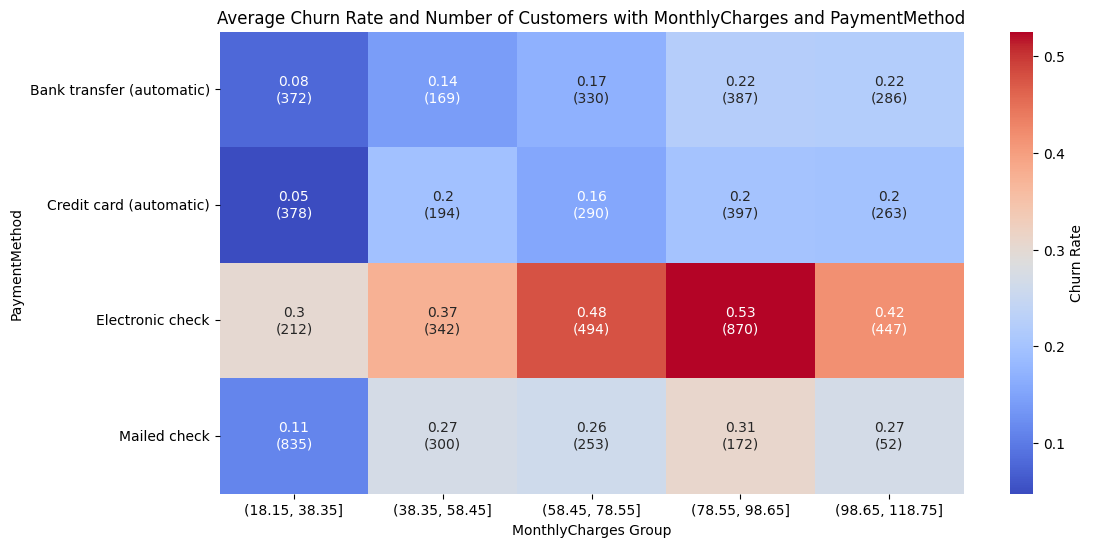
plt.title('Average Churn Rate and Number of Customers with MonthlyCharges and PaymentMethod')

plt.xlabel('MonthlyCharges Group')

plt.ylabel('PaymentMethod')

plt.savefig('../results/plots/Churn\_rate\_heatmap.png', bbox\_inches='tight')

plt.show()



***Fig: Churn Rate Heatmap***

* Distribution of data by Churn in numerical columns.

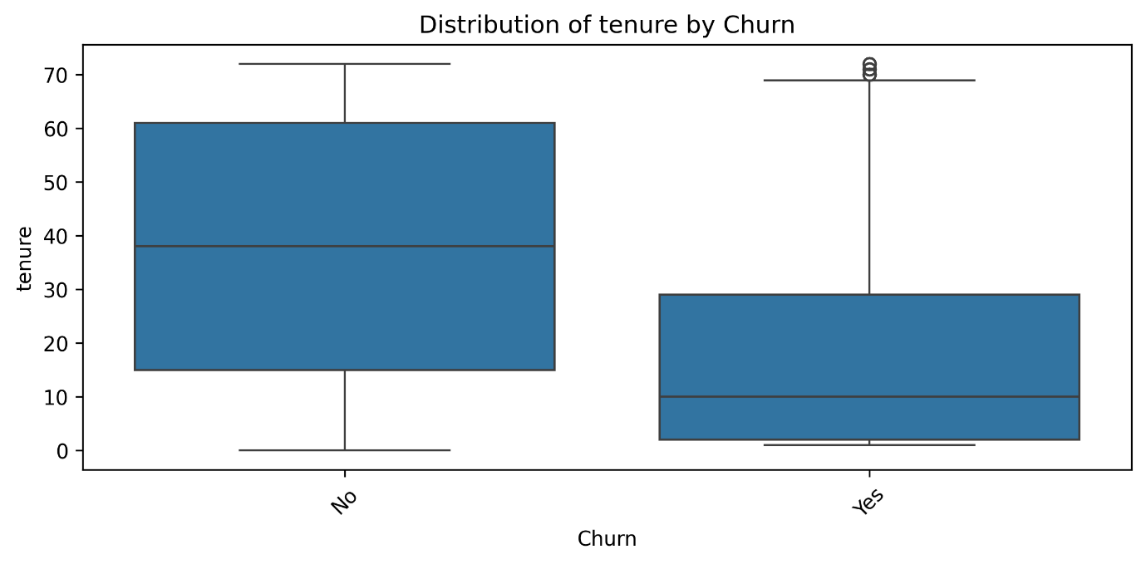
for col **in** numerical\_cols:

plt.figure(figsize=(8, 4))

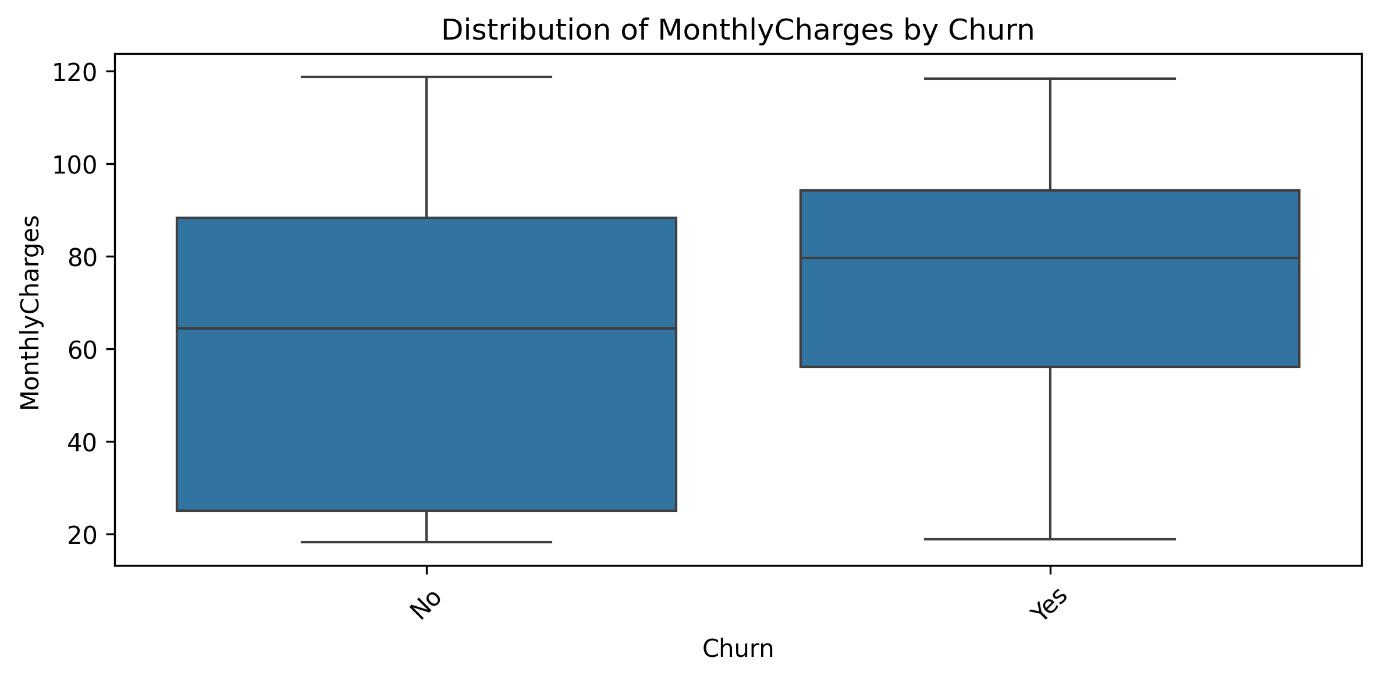
sns.boxplot(x=target, y=col, data=df)

plt.title(f" Distribution of **{**col**}** variable by Churn")

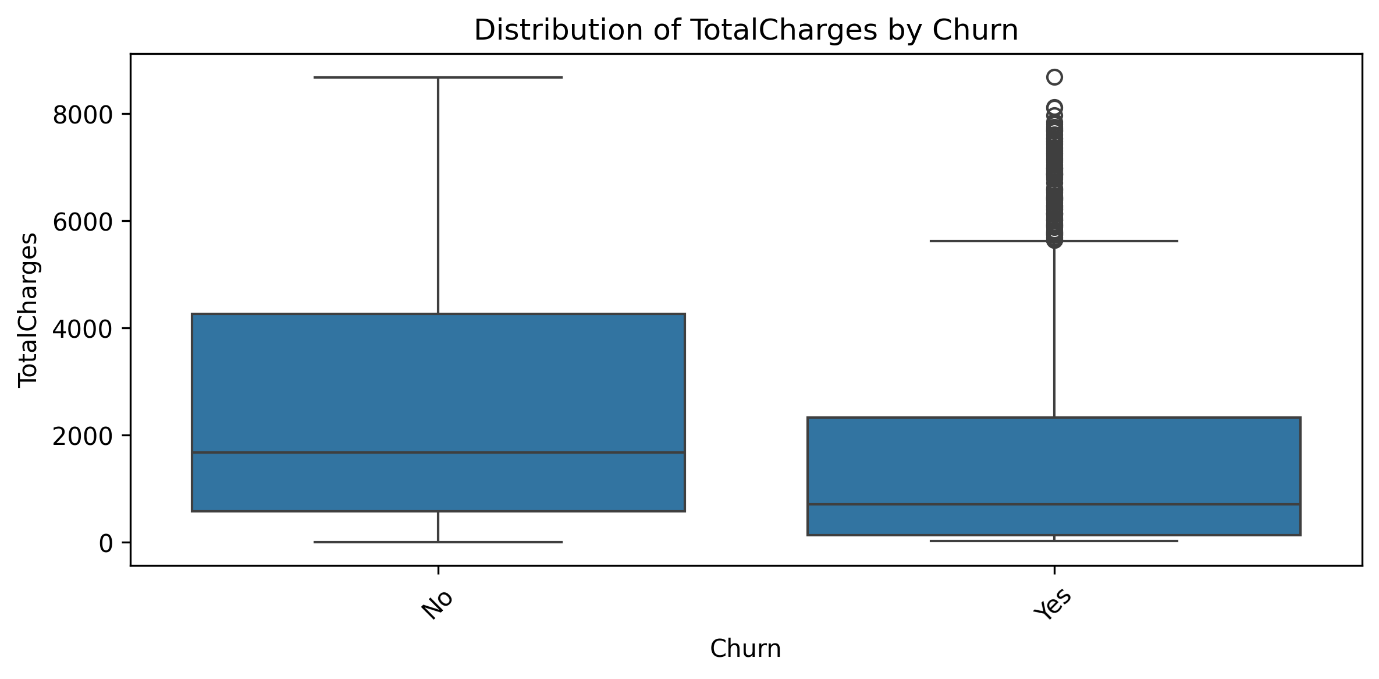
plt.show()



***Fig: Distribution of tenure by Churn***

******

***Fig: Distribution of MonthlyCharges by Churn***

******

***Fig: Distribution of TotalCharges by Churn***

**3.3 Feature Engineering & Data Preprocessing**

**1. Feature Engineering**

* We created a "High-Risk Customers" Feature by flagging customers who are likely to churn base on these conditions (payment method, short tenure duration, and high monthly charge rate).

class HighRiskFeatureGenerator(BaseEstimator, TransformerMixin):

    def \_\_init\_\_(self):

        self.train\_80th\_ = None

    def fit(self, X, y=None):

        # Calculate percentile from TRAINING DATA only

        self.train\_80th\_ = X['MonthlyCharges'].quantile(0.8)

        return self

    def transform(self, X):

        X = X.copy()

        X['HighRiskCustomers'] = (

            (X['PaymentMethod'] == 'Electronic check') &

            (X['MonthlyCharges'] > self.train\_80th\_)

        ).astype(int)

        return X

**2. Encoding**

* Since we are doing Churn Analysis, the target ‘Churn’ column needed to be converted to binary values.

df['Churn'] = df['Churn'].map({'Yes': 1, 'No': 0})

* Split the target and feature columns, then split the data for training and testing

X = df.drop(columns=['Churn'])

y = df['Churn']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

    X, y,

    test\_size=0.2,

    random\_state=42,

    stratify=y

)

* The missing values are handled, the numerical columns are scaled, and categorical columns are one-hot encoded in the model pipeline setup.
* Imbalance Class problem in the data is also addressed using SMOTE ensemble method.

pipeline = Pipeline([

    ('imputer', DataFrameImputer(numerical\_cols, strategy='median')),

    ('feature\_engineer', HighRiskFeatureGenerator()),

    ('preprocessor', ColumnTransformer([

        ('num', StandardScaler(), numerical\_cols),

        ('cat', OneHotEncoder(handle\_unknown='ignore'), categorical\_cols)

    ], remainder='passthrough')),

    ('smote', SMOTE(random\_state=42)),

    ('classifier', XGBClassifier())

])

**4. Model Selection - XGBoost**

**4.1 Model Introduction**

In this project, XGBoost (Extreme Gradient Boosting) was selected for predicting telecom customer churn due to its proven effectiveness in handling structured tabular data, a common format for customer behavior datasets. Customer churn prediction is a **binary classification problem** (churn vs. non-churn) that often involves complex interactions between features (e.g., usage patterns, billing history, demographics). XGBoost excels in such scenarios because:

* It combines **gradient boosting** with **regularization** (L1/L2) to balance model complexity and generalization, critical for noisy telecom datasets.
* It automatically handles **missing values**, reducing preprocessing effort.
* It supports **class imbalance** through the scale\_pos\_weight parameter, which adjusts for skewed churn rates (e.g., 5% churn vs. 95% non-churn).
* Its **built-in functions (**eg. **cross-validation** and **early stopping***)* prevent overfitting, ensuring robustness even with limited training data.

**4.2 Advantages Over Other Models**

XGBoost outperforms traditional and ensemble models in accuracy and efficiency for churn prediction:

|  |  |  |
| --- | --- | --- |
| **Model** | **Limitations** | **XGBoost Advantages** |
| **Logistic Regression** | Linear decision boundaries fail to capture non-linear relationships (e.g., usage spikes preceding churn). | Handles non-linear patterns via sequential decision trees and feature splits. |
| **Random Forest** | Prone to overfitting on noisy data; slower inference with large forests. | Optimized with gradient boosting (corrects errors iteratively) and regularization for tighter control. |
| **SVM** | Computationally expensive for large datasets; struggles with class imbalance. | Faster training via parallel processing; handles imbalance via ***scale\_pos\_weight.*** |
| **Neural Networks** | Requires massive data and tuning; lacks interpretability. | More efficient on smaller datasets; provides feature importance scores for business insights. |

**Example**: Telecom datasets often include categorical features like *contract type* (monthly, annual) or *payment method* (credit card, bank transfer). XGBoost’s experimental ***enable\_categorical=True*** (for Pandas category dtype) simplifies encoding, unlike one-hot encoding in logistic regression, which inflates dimensionality.

**4.3 Training the Model**

* Model initiating and training

# Initialize and train model

pipeline.fit(X\_train, y\_train)

* Hyperparameter tuning using GridSearchCV so that the model can perform its best.

new\_pipeline = clone(pipeline)

scale\_pos\_weight\_resampled = 1.0

cv = StratifiedShuffleSplit(

    n\_splits=1,

    test\_size=0.2,

    random\_state=42) # Define grid

param\_grid = {

    'classifier\_\_n\_estimators': [100, 200],

    'classifier\_\_max\_depth': [3, 5],

    'classifier\_\_learning\_rate': [0.05, 0.1],

    'classifier\_\_subsample': [0.8, 1.0],

    'classifier\_\_colsample\_bytree': [0.8, 1.0],

    'classifier\_\_gamma': [0, 0.2],

    'classifier\_\_reg\_alpha': [0, 0.5],

    'classifier\_\_scale\_pos\_weight': [1]  # SMOTE balances classes

# Initialize GridSearchCV

grid\_search = GridSearchCV(

    estimator=new\_pipeline,

    param\_grid=param\_grid,

    scoring='roc\_auc',

    cv=cv,

    verbose=2,

    n\_jobs=-1)

# Execute grid search

grid\_search.fit(X\_train, y\_train)

**5. Evaluation Results**

**5.1 XGBoost Before Tuning**

**Classification Report -**

Classification Report (AUC-ROC = 0.821):

precision recall f1-score support

0 0.85 0.84 0.85 1035

1 0.58 0.60 0.59 374

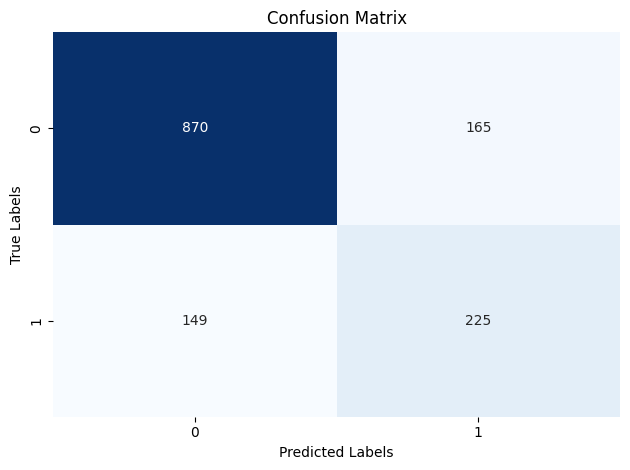
accuracy 0.78 1409

macro avg 0.72 0.72 0.72 1409

weighted avg 0.78 0.78 0.78 1409

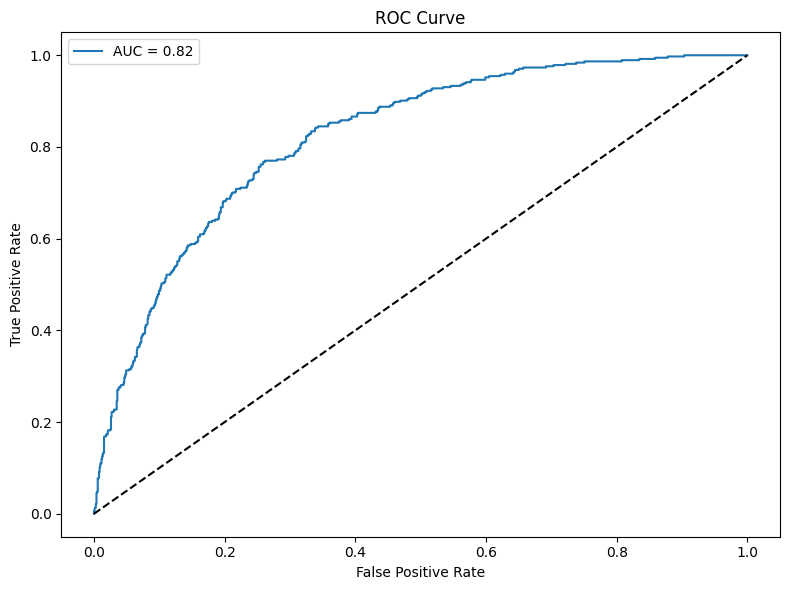
Accuracy: 0.777

**Confusion Matrix**

****

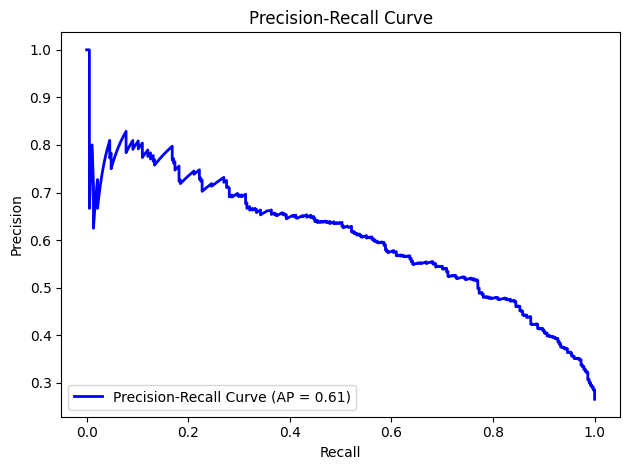
***Fig: XGBoost Confusion Matrix Before Tuning***

**Receiver Operating Characteristic curve (ROC Curve)**

****

***Fig: XGBoost ROC Curve Before Tuning***

**Precision Recall Curve**

****

***Fig: XGBoost Precision Recall Curve Before Tuning***

**5.2 XGBoost After Tuning**

**Classification Report –**

Classification Report (AUC-ROC = 0.843):

precision recall f1-score support

0 0.88 0.80 0.84 1035

1 0.56 0.71 0.62 374

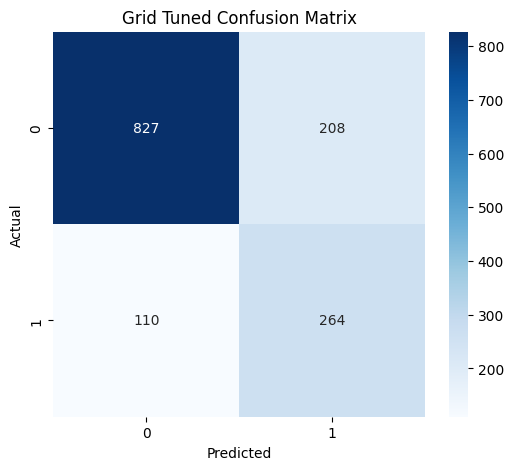
accuracy 0.77 1409

macro avg 0.72 0.75 0.73 1409

weighted avg 0.80 0.77 0.78 1409

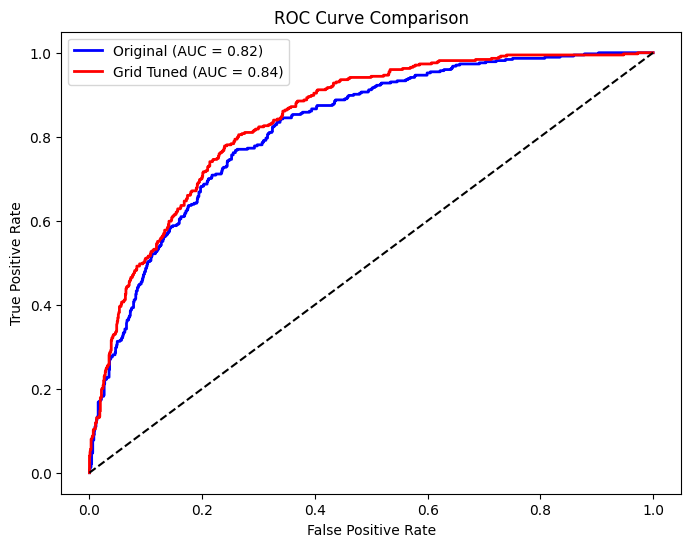
Accuracy: 0.774

**Confusion Matrix**

****

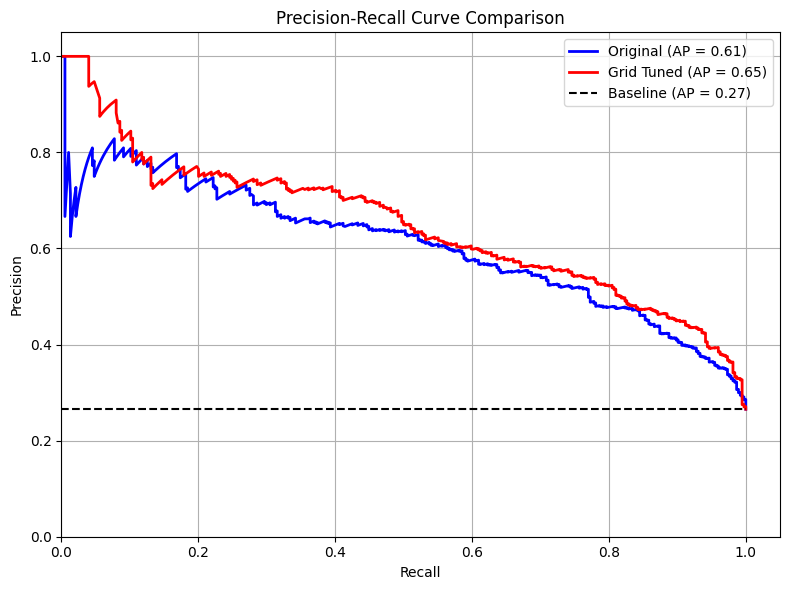
***Fig: XGBoost Confusion Matrix After Tuning***

**ROC Curve Comparison After Tuning**

****

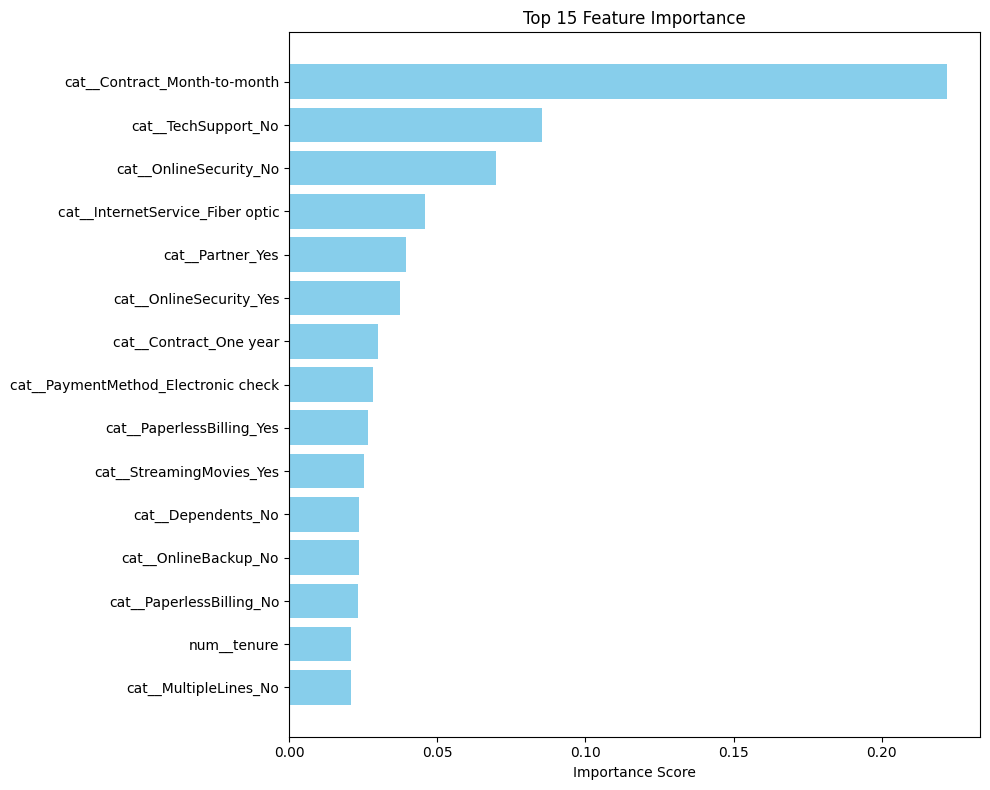
***Fig: XGBoost ROC Curve Comaprison After Tuning***

**Precision Recall Curve Comparison After Tuning**

****

***Fig: XGBoost Precision Recall Curve Comparison After Tuning***

**5.3 XGBoost Feature Importance**

****

***Fig: XGBoost Feature Importance***

**5.4** **Performance Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Before Tuning** | **After Tuning** | **Change** |
| **AUC-ROC** | 0.821 | 0.843 | +2.2% |
| **Class 1 Recall** | 0.60 | 0.71 | +11.0% |
| **Class 1 F1-Score** | 0.59 | 0.62 | +5.1% |
| **Macro Avg F1** | 0.72 | 0.73 | +1.4% |
| **Weighted Avg F1** | 0.78 | 0.78 | No Change |
| **Accuracy** | 0.777 | 0.774 | -0.3% |

The hyperparameter tuning successfully improved the model’s ability to detect the minority class (**Class 1**) while maintaining overall accuracy. The trade-offs align with typical imbalance mitigation strategies, but further refinement could optimize precision for critical use cases. The **AUC-ROC improvement** validates the tuning strategy, though domain-specific costs should guide next steps.

**6. Conclusion**

Predicting customer churn is a critical challenge for telecom companies, where retaining customers directly impacts profitability and long-term sustainability. This project aimed to address this challenge by developing a machine learning model capable of identifying at-risk customers early, enabling proactive retention strategies. Through systematic data preparation, model selection, and hyperparameter tuning, we successfully built an XGBoost-based churn prediction system with actionable insights for telecom businesses.

In conclusion, this project demonstrates the power of machine learning in transforming customer churn prediction from a reactive to a strategic business tool. By combining technical rigor with business context, telecom companies can turn data into actionable strategies, fostering customer loyalty and driving sustainable growth.