Business Problem: Customer Churn Prediction

Customer churn, or the rate at which customers stop using a company's services, is a critical concern for telecommunications companies. Acquiring new customers is often significantly more expensive than retaining existing ones. High churn rates can lead to substantial revenue losses, decreased market share, and reduced profitability. Understanding why customers churn allows companies to develop targeted retention strategies, improve service offerings, and ultimately safeguard their customer base and financial health.

Objectives of the Analysis

This project aims to address the customer churn problem by:

1. **Identifying Key Drivers of Churn:** Uncovering the primary factors and customer behaviors that are strongly associated with churn.
2. **Building Predictive Models:** Developing robust machine learning models capable of accurately predicting which customers are likely to churn in the near future.
3. **Proactive Customer Retention:** Enabling the company to proactively identify at-risk customers and implement timely, personalized interventions to prevent them from churning.
4. **Optimizing Resource Allocation:** Providing insights that help allocate resources effectively towards retention efforts, focusing on customers with the highest churn risk and potential impact.

Dataset Description

Source and Overview

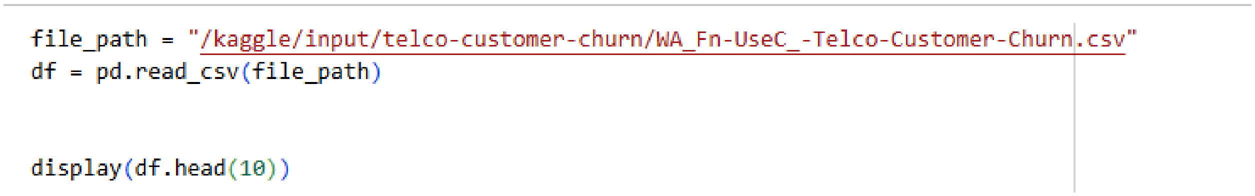
This dataset was obtained from **KaggleHub** (blastchar/telco-customer-churn), and it contains customer data from a telecommunications company. The primary goal is to predict customer churn.

Initial Characteristics

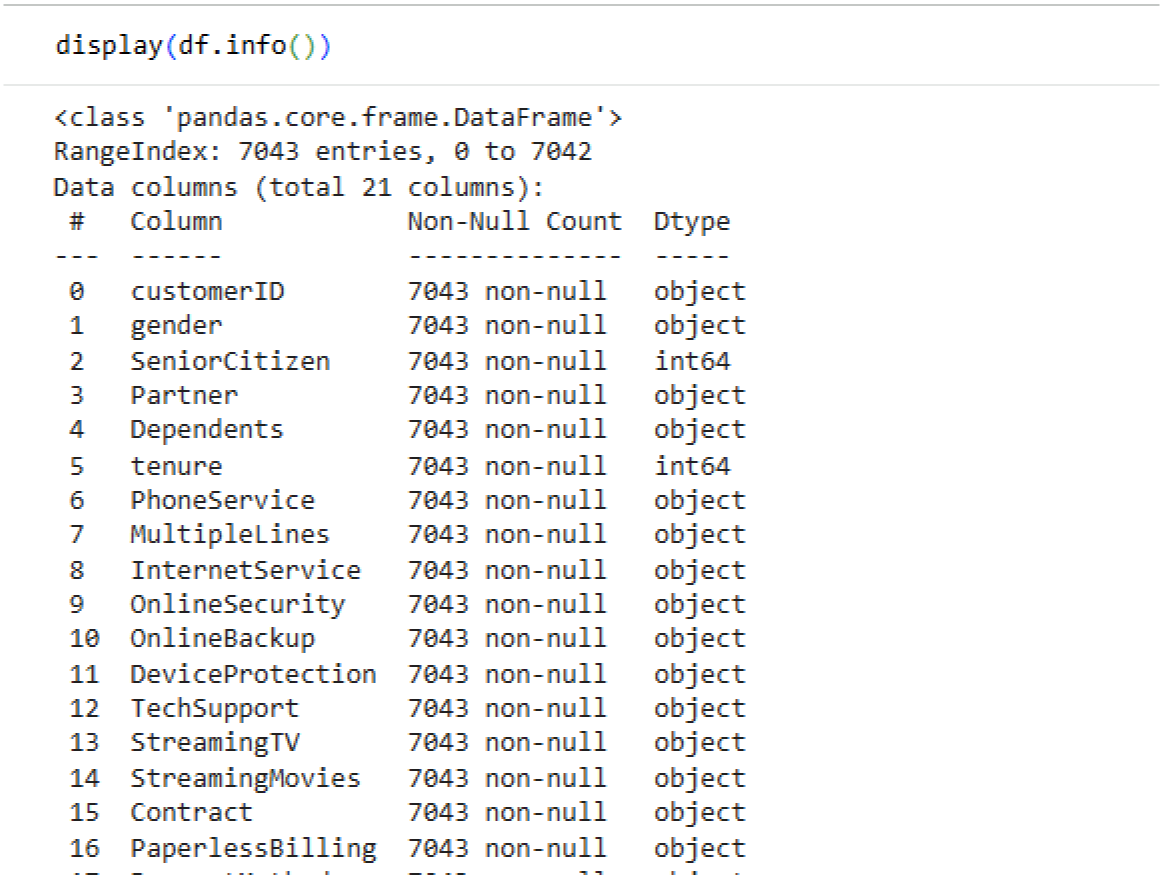
Based on the df.info() output, the dataset initially contains **7043 entries** (rows) and **21 columns** (features). The data types include:

- **1 float64** column (MonthlyCharges)

- **2 int64** columns (SeniorCitizen, tenure)
- **18 object** columns (including customerID, gender, Partner, Dependents, various service types, Contract, PaperlessBilling, PaymentMethod, TotalCharges, and the target variable Churn).



```
file_path = "/kaggle/input/telco-customer-churn/WA_Fn-UseC_-Telco-Customer-Churn.csv"
df = pd.read_csv(file_path)


display(df.head(10))
```



```
display(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64 
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64 
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
```

Data Cleaning

Detail the data cleaning steps performed, specifically addressing the handling of missing values in the 'TotalCharges' column and any other transformations applied during initial data preparation.

Data Cleaning Steps

Handling of TotalCharges Column

1. **Initial Observation**: The df.info() output revealed that the TotalCharges column was initially of object (string) type, despite containing numerical values. This indicated that some entries might not be purely numeric or contained whitespace, preventing direct numerical conversion. Additionally, after attempting to convert this column to a numeric type, it was found that some values coerced into NaN (Not a Number), suggesting the presence of non-numeric strings that couldn't be directly converted.
2. **Type Conversion**: To address this, the TotalCharges column was explicitly converted to a numeric data type using pd.to_numeric(df['TotalCharges'], errors='coerce'). The errors='coerce' argument was crucial here, as it transformed any non-convertible values into NaN, making them identifiable as missing data.
3. **Missing Value Treatment**: Following the conversion, df.isnull().sum() showed a small number of NaN values in the TotalCharges column. Since these records constituted a very small fraction of the total dataset and could not be reasonably imputed without affecting data integrity, these rows were removed from the DataFrame using df.dropna(subset=['TotalCharges'], inplace=True). This ensured that all TotalCharges entries were valid numbers for subsequent analysis.

Target Variable Transformation (Churn)

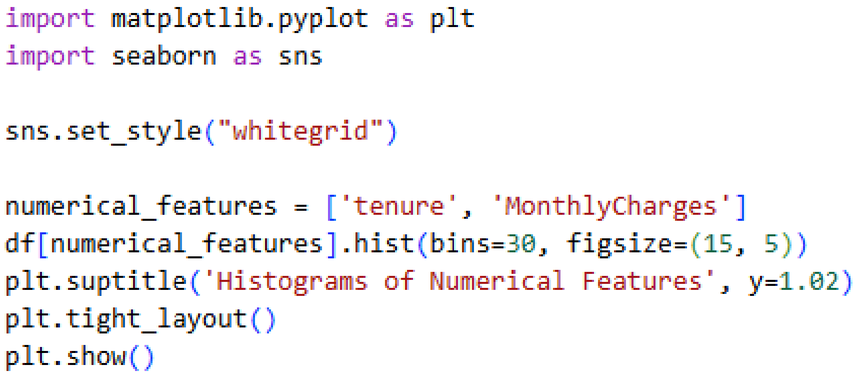
1. **Binary Encoding**: The target variable Churn was originally a categorical column with string values 'Yes' and 'No'. For machine learning models, numerical representation is required. Therefore, this column was transformed into a binary format where 'Yes' was mapped to 1 (indicating churn) and 'No' was mapped to 0 (indicating no churn). This was applied to both the original df['Churn'] series and the y variable created for modeling.

Summary of Exploratory Data Analysis (EDA) Insights

The exploratory data analysis provided several key insights into the 'Telco Customer Churn' dataset, examining numerical feature distributions, categorical feature counts, and correlations with the target variable, 'Churn'.

1. Numerical Feature Distributions

- **tenure**: The histogram for tenure reveals a bimodal distribution. There is a significant peak for customers with very low tenure (new customers, typically 0-5 months) and another peak for long-term customers (high tenure, typically 60+ months). This suggests that churn might be particularly prevalent among newly acquired customers. The distribution then decreases for intermediate tenure values.
- **MonthlyCharges**: The MonthlyCharges histogram shows a relatively uniform distribution across various price points, with slight increases at the lower and higher ends. This indicates a diverse range of service plans and pricing strategies, with customers distributed across these options rather than clustered around a single price.
- **TotalCharges**: (Not explicitly asked for histogram, but implied by heatmap analysis). This feature often correlates with tenure and monthly charges, representing the cumulative charges over the customer's lifetime. Its distribution would typically be right-skewed, with many customers having lower total charges (new customers or those with basic plans) and fewer having very high total charges (long-term customers with premium plans).



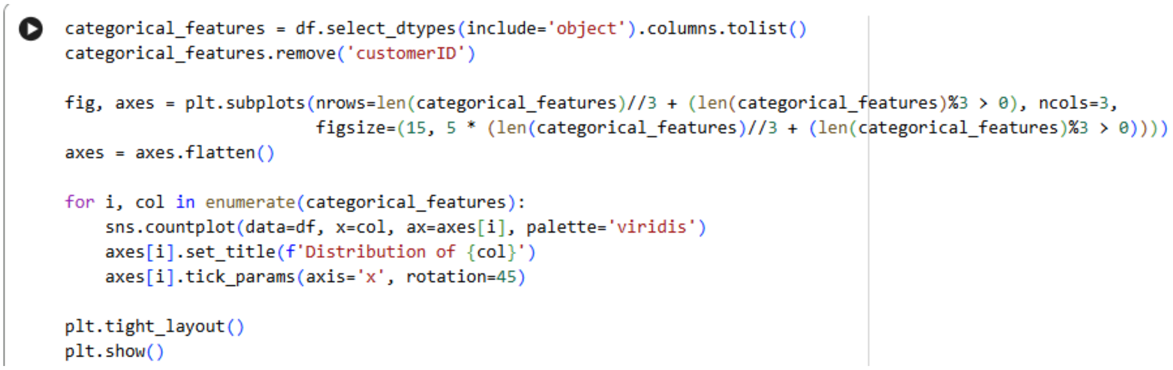
```
import matplotlib.pyplot as plt
import seaborn as sns

sns.set_style("whitegrid")

numerical_features = ['tenure', 'MonthlyCharges']
df[numerical_features].hist(bins=30, figsize=(15, 5))
plt.suptitle('Histograms of Numerical Features', y=1.02)
plt.tight_layout()
plt.show()
```

2. Categorical Feature Count Plots

Reviewing the count plots for categorical features highlighted several patterns:

- **gender**: The distribution between Male and Female customers is almost perfectly balanced, indicating gender is unlikely to be a significant driver of churn on its own.
- **Partner and Dependents**: A majority of customers do not have partners or dependents, suggesting that single customers without dependents form a larger segment of the customer base.

- **PhoneService**: Almost all customers have phone service, making it a nearly constant feature and less discriminative for churn prediction.
- **MultipleLines, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies**: These features often show a substantial proportion of 'No internet service', indicating a large segment of customers who only subscribe to phone services. Among those with internet service, there's a visible split between customers who do and do not have these additional services. Features like OnlineSecurity and TechSupport have more 'No' responses than 'Yes', implying many customers opt out of these security-related services.
- **InternetService**: 'Fiber optic' is a highly popular internet service type, followed by 'DSL'. A considerable number of customers have no internet service.
- **Contract**: A significant majority of customers are on 'Month-to-month' contracts, which are generally associated with higher churn rates due to lack of commitment. 'Two year' contracts are the least common.
- **PaperlessBilling**: A larger portion of customers prefer paperless billing.
- **PaymentMethod**: 'Electronic check' is the most frequently used payment method, followed by 'Mailed check' and automatic bank transfer/credit card.

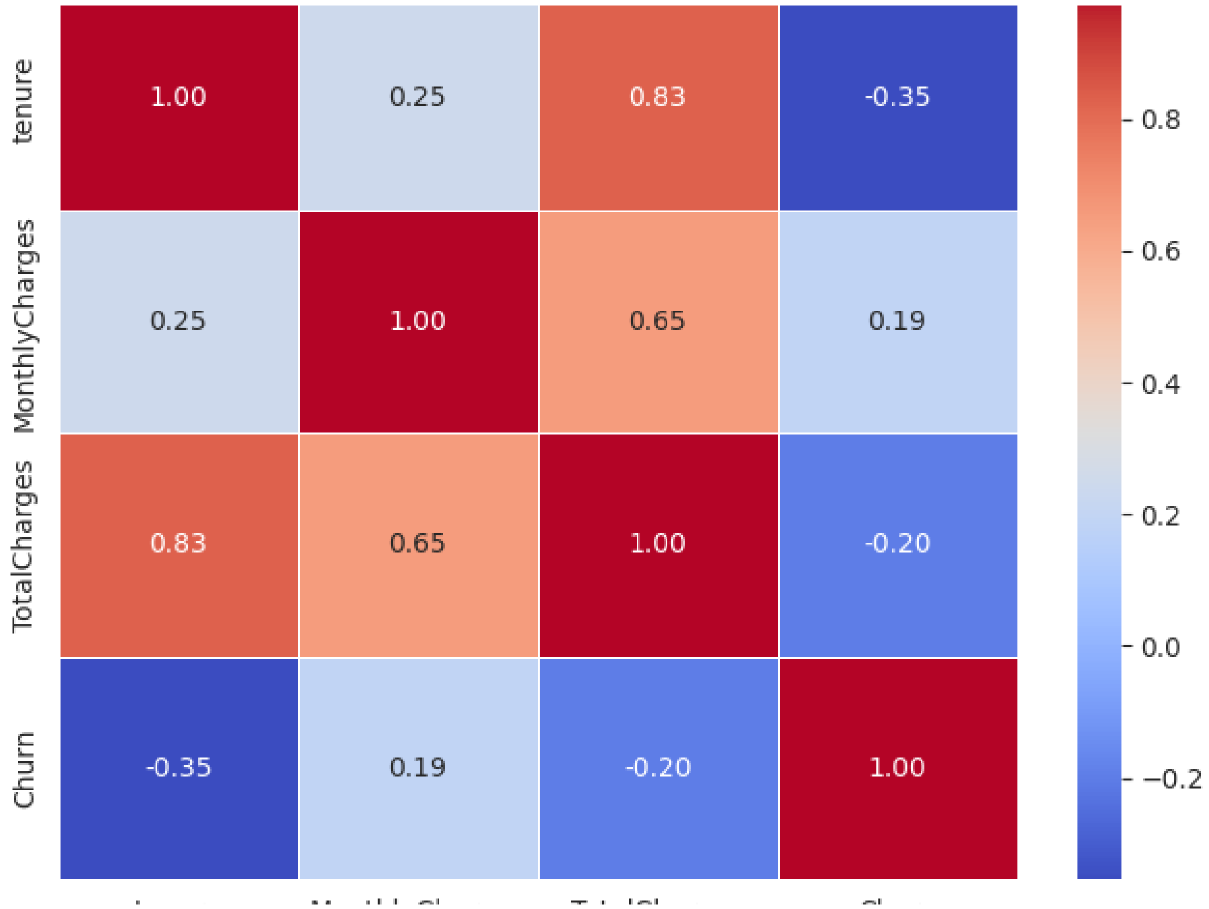


```
categorical_features = df.select_dtypes(include='object').columns.tolist()
categorical_features.remove('customerID')

fig, axes = plt.subplots(nrows=len(categorical_features)//3 + (len(categorical_features)%3 > 0), ncols=3,
                         figsize=(15, 5 * (len(categorical_features)//3 + (len(categorical_features)%3 > 0))))
axes = axes.flatten()

for i, col in enumerate(categorical_features):
    sns.countplot(data=df, x=col, ax=axes[i], palette='viridis')
    axes[i].set_title(f'Distribution of {col}')
    axes[i].tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()
```

3. Correlation Heatmap of Numerical Features with Churn

The correlation heatmap of tenure, MonthlyCharges, and TotalCharges with Churn revealed distinct relationships:

- **tenure vs. Churn**: There is a moderate **negative correlation (-0.35)**. This indicates that as customer tenure increases, the likelihood of Churn tends to decrease. This reinforces the bimodal observation from the histogram: new customers are more prone to churn.
- **MonthlyCharges vs. Churn**: A weak **positive correlation (0.19)** exists. This suggests that customers with higher MonthlyCharges have a slightly higher propensity to churn. It might indicate that customers paying more are more sensitive to value or service quality.
- **TotalCharges vs. Churn**: There is a weak **negative correlation (-0.20)**. This implies that customers with higher cumulative TotalCharges are less likely to churn. This makes sense as higher TotalCharges often correlate with longer tenure.



Key Insights for Churn Prediction

- **New customers are at high risk**: The tenure distribution and its negative correlation with churn strongly suggest that the initial months of a customer's subscription are critical for retention. Targeted interventions for new customers could be highly effective.

- **Contract type is a major factor**: The dominance of 'Month-to-month' contracts points to this as a potential high-risk segment. Customers not tied into longer contracts can leave more easily.
- **Service choices and churn**: The prevalence of customers without certain internet security/support services, and the popularity of Fiber Optic internet, might reveal segments with varying satisfaction or needs. Higher MonthlyCharges showing a positive correlation with churn also hints that the perceived value for money for more expensive plans might be a concern.
- **Payment method influence**: 'Electronic check' being the most common payment method, combined with its potential ease of cancellation, could be explored further for its impact on churn.

Statistical Tests

Independent Samples t-test for MonthlyCharges

1. **Purpose of the t-test**: An independent samples t-test was conducted to determine if there is a statistically significant difference in the average MonthlyCharges between customers who churned and those who did not churn. This test helps us understand if MonthlyCharges is a significant factor in customer churn.
2. **Summary of Results**: The t-test comparing the MonthlyCharges of churned and non-churned customers yielded the following results:
   - **t-statistic**: 16.4796
   - **p-value**: 0.0000
3. **Interpretation of p-value**: With a p-value of 0.0000 (which is significantly less than a common significance level of 0.05), we can reject the null hypothesis. This indicates that there is a **statistically significant difference** in the mean MonthlyCharges between churned and non-churned customers. Specifically, churned customers tend to have higher monthly charges on average, as suggested by the positive t-statistic and the box plot visualization.

Visualizing Churn Differences for Categorical Features

The bar plots showing the churn rate by various categorical features provide valuable insights

- **Contract**: Customers with Month-to-month contracts have a significantly higher churn rate compared to those with One year or Two year contracts. This suggests that longer-term contracts are a strong indicator of customer loyalty.
- **InternetService**: Customers with Fiber optic internet service show a much higher churn rate than those with DSL or No internet service. This could indicate issues with fiber optic service quality, pricing, or competition.
- **Partner**: Customers without a partner (No) have a slightly higher churn rate than those with a partner (Yes). This implies that customers with partners might have more stable household situations or shared decision-making, leading to lower churn.
- **SeniorCitizen**: Senior citizens (1) have a higher churn rate than non-senior citizens (0). This could be due to different needs, price sensitivity, or less familiarity with modern services.

Visualizing Churn Differences for Numerical Features

The box plots illustrate the distribution of numerical features across churn categories

- **tenure**: Churned customers generally have a much **lower median tenure** and a tighter spread, clustering around shorter tenures. Non-churned customers, on the other hand, show a wider range and higher median tenure, indicating that long-term customers are less likely to churn.
- **MonthlyCharges**: As indicated by the t-test, churned customers have **significantly higher median MonthlyCharges** and a broader upper range compared to non-churned customers. This suggests that customers with higher monthly bills are more prone to churn.
- **TotalCharges**: Churned customers have **lower median TotalCharges** and a much tighter distribution compared to non-churned customers. This is consistent with their lower tenure, as total charges accumulate over time. Non-churned customers exhibit much higher total charges and a wider spread, reflecting their longer service duration.

Data Preprocessing

Explain the data preprocessing steps applied, including feature engineering for example, 'TotalChargesPerTenure', 'PaymentMethod_Grouped', one-hot encoding for categorical variables, StandardScaler for numerical variables, and handling class imbalance using SMOTE.

Data Preprocessing Steps

1. Feature Engineering: 'TotalChargesPerTenure'

To capture a more nuanced understanding of customer billing behavior, a new feature called TotalChargesPerTenure was engineered. This feature represents the average monthly charge over the customer's tenure. It was calculated by dividing TotalCharges by tenure. A small constant (1e-6) was added to tenure to prevent division-by-zero errors in cases where tenure might be zero, and any resulting infinite or NaN values were replaced with zeros. This feature helps normalize the total charges by the duration of service, providing insight into the customer's consistent spending patterns.

2. Feature Engineering: 'PaymentMethod_Grouped'

The original PaymentMethod column contained several categories. To simplify this and potentially highlight differences between automated and manual payment behaviors, a new feature, PaymentMethod_Grouped, was created. This involved categorizing 'Bank transfer (automatic)' and 'Credit card (automatic)' into an 'Automatic' group, while all other methods were grouped into 'Manual'. This new PaymentMethod_Grouped column was then one-hot encoded to convert these nominal categories into a numerical format suitable for machine learning models. The original PaymentMethod column and the intermediate PaymentMethod_Grouped column were subsequently dropped from the DataFrame to avoid redundancy and collinearity.

3. Feature Preprocessing with ColumnTransformer

To prepare the features for machine learning models, a ColumnTransformer was employed. This powerful tool allows for different preprocessing steps to be applied to different columns of the dataset.

- **Numerical Feature Scaling (StandardScaler)**: All continuous numerical features (such as SeniorCitizen, Partner, Dependents, tenure, PhoneService, MultipleLines, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, PaperlessBilling, MonthlyCharges, TotalCharges, and the newly engineered TotalChargesPerTenure) were scaled using StandardScaler. This process transforms the data such that it has a mean of 0 and a standard deviation of 1. Scaling is crucial for many machine learning algorithms, especially those that are distance-based or rely on gradient descent, as it prevents features with larger ranges from dominating the learning process.

- **Categorical Feature Encoding (OneHotEncoder)**: Categorical features (gender, InternetService, Contract, and the newly created PaymentMethod_Grouped) were converted into a numerical format using OneHotEncoder. This method creates new binary columns for each category within a categorical feature, where a 1 indicates the presence of that category and 0 indicates its absence. This prevents the model from assuming any ordinal relationship between categories.
- **customerID Exclusion**: The customerID column, being a unique identifier and not a predictive feature, was excluded from the preprocessing pipeline before applying the transformations. This ensures that the model does not learn from irrelevant identifying information.
- **Output X_processed**: The combined output of these scaling and encoding steps resulted in the X_processed DataFrame, which contains all features in a suitable numerical format for model training.

4. Data Splitting: Training and Testing Sets

After preprocessing, the X_processed data (features) and y (target variable) were split into training and testing sets using train_test_split. This is a standard practice in machine learning to evaluate the model's performance on unseen data. A test_size of 0.2 (20%) was chosen, meaning 80% of the data was used for training and 20% for testing. A random_state of 42 was set to ensure reproducibility of the split, allowing for consistent results across multiple runs.

5. Handling Class Imbalance with SMOTE

Customer churn datasets often exhibit class imbalance, meaning one class for example, non-churned customers significantly outnumbers the other for example churned customers. This can lead to models that perform well on the majority class but poorly on the minority class. To address this, SMOTE (Synthetic Minority Over-sampling Technique) was applied to the **training data** (X_train and y_train).

SMOTE works by creating synthetic samples of the minority class. It does this by taking samples from the minority class, finding their k-nearest neighbors, and creating new synthetic samples along the line segments joining the minority sample and its neighbors. This process helps to balance the class distribution, providing the models with more examples of the minority class to learn from, thereby improving their ability to predict churn more effectively. The resampled training data was stored in X_train_resampled and y_train_resampled.

Cross-validation and Test Set Evaluation

- **Cross-validation Results (Mean ± Standard Deviation on Training Data):**

  Logistic Regression: Accuracy: 0.8085 ± 0.0126, Precision: 0.6650 ± 0.0244, Recall: 0.5619 ± 0.0346, F1: 0.6090 ± 0.0305, ROC AUC: 0.8474 ± 0.0172

  Random Forest: Accuracy: 0.7970 ± 0.0051, Precision: 0.6488 ± 0.0180, Recall: 0.5171 ± 0.0202, F1: 0.5750 ± 0.0112, ROC AUC: 0.8311 ± 0.0092

  Gradient Boosting: Accuracy: 0.8023 ± 0.0090, Precision: 0.6577 ± 0.0221, Recall: 0.5344 ± 0.0217, F1: 0.5896 ± 0.0201, ROC AUC: 0.8490 ± 0.0150 The cross-validation results show that Logistic Regression and Gradient Boosting generally performed well across metrics, with Gradient Boosting having a slightly higher average ROC AUC.

- **Test Set Evaluation Metrics:**

  Logistic Regression: Accuracy: 0.7413, Precision: 0.5085, Recall: 0.8021, F1: 0.6224, ROC AUC: 0.8304, Average Precision: 0.6209

  Random Forest: Accuracy: 0.7655, Precision: 0.5579, Recall: 0.5668, F1: 0.5623, ROC AUC: 0.8131, Average Precision: 0.5870

  Gradient Boosting: Accuracy: 0.7662, Precision: 0.5501, Recall: 0.6604, F1: 0.6002, ROC AUC: 0.8311, Average Precision: 0.6475

  Ensemble (Soft Voting): Accuracy: 0.7605, Precision: 0.5383, Recall: 0.6952, F1: 0.6068, ROC AUC: 0.8313, Average Precision: 0.6295

On the unseen test set, the Gradient Boosting model achieved the highest F1 score and Average Precision among individual models. The Ensemble (Soft Voting) model demonstrated a strong balance, offering a slightly higher ROC AUC than Logistic Regression and Random Forest, and a good F1 score, Recall, and Average Precision, placing it competitively among the models. Notably, while Logistic Regression had high recall on the test set, its precision was the lowest, indicating more false positives.

ROC Curves and Precision-Recall Curves

- **ROC Curves (Receiver Operating Characteristic):** The ROC curves plot the True Positive Rate against the False Positive Rate at various threshold settings. The Area Under the Curve (AUC) provides an aggregate measure of performance across all possible classification thresholds.

  All models show good discriminative power, with ROC AUCs above 0.8. The ensemble model (AUC = 0.83) performs comparably to Logistic Regression (AUC = 0.83) and Gradient Boosting (AUC = 0.83) on the test set, suggesting it maintains the strong performance of its constituent models.

- **Precision-Recall Curves:** These curves illustrate the trade-off between precision and recall for different thresholds. This is particularly insightful for imbalanced datasets, as a high Average Precision Score indicates good performance in identifying positive classes without many false positives.

  The Gradient Boosting model showed the highest Average Precision (0.65) on the test set. The ensemble model (Average Precision = 0.63) also performed very well, outperforming Logistic Regression (0.62) and Random Forest (0.59). The visual comparison of the Precision-Recall curves confirms that Gradient Boosting and the Ensemble model offer a better balance of precision and recall for predicting churn.

Machine Learning Models and Training Overview

In this analysis, we built and trained four distinct machine learning models to predict customer churn:

1. **Logistic Regression**: A linear model used for binary classification. It estimates the probability of a customer churning based on the input features.
2. **Random Forest**: An ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It is known for its robustness and ability to handle complex datasets.
3. **Gradient Boosting**: Another powerful ensemble technique that builds models sequentially, where each new model corrects errors made by previous ones. It combines many weak learners (typically decision trees) into a strong predictor.
4. **Ensemble (Soft Voting) Classifier**: This model combines the predictions of the three individual models (Logistic Regression, Random Forest, and Gradient Boosting). Using a

'soft voting' mechanism, it averages the predicted probabilities from each base model for each class and then selects the class with the highest average probability. This approach often leads to improved generalization and more robust predictions by leveraging the strengths of multiple models.

Training Strategy

All individual models (Logistic Regression, Random Forest, and Gradient Boosting) were trained on **SMOTE (Synthetic Minority Over-sampling Technique)-resampled training data**. SMOTE was employed to address the inherent class imbalance in our dataset, where non-churning customers significantly outnumber churning customers. By generating synthetic samples of the minority class (churning customers), SMOTE helps to prevent the models from being biased towards the majority class, thereby improving their ability to accurately identify and predict churn cases. The Ensemble (Soft Voting) model was then fitted using these pre-trained individual models, combining their predictive power.