Case Study: Predicting Customer Churn for a Telecom Company

---

1. Introduction

The telecom industry faces a challenge in customer retention. Predicting whether a customer will churn (leave the service) is crucial for businesses to take proactive measures and reduce customer attrition. In this case study, we aim to build a predictive model that uses customer data to predict whether a customer will churn or not. By leveraging machine learning algorithms such as K-Nearest Neighbors, Naïve Bayes, Decision Trees, Random Forest, AdaBoost, XGBoost, and Support Vector Machine (SVM), we will predict the churn status of customers based on their demographic and service usage data.

---

2. Problem Statement

The goal of this case study is to predict whether a customer will churn (leave the service) based on their characteristics. This is a binary classification problem where we need to predict the target variable Churn (1 = Churn, 0 = No Churn) using features such as customer age, service plan, monthly usage, and payment method. Accurate churn prediction will allow the telecom company to focus on high-risk customers and deploy targeted retention strategies.

---

3. Dataset Overview

The dataset contains the following columns:

Column Name	Description	Type	Range
Age	Age of the customer in years	Numerical	Integer, between 18 and 70
Tenure	The number of months the customer has been with the company	Numerical	Integer, between 1 and 72
ServicePlan	The service plan chosen by the customer (e.g., Basic, Standard, Premium)	Categorical	"Basic", "Standard", "Premium"
PaymentMethod	The method used to pay for the service (e.g., CreditCard, BankTransfer, Cash)	Categorical	"CreditCard", "BankTransfer", "Cash"
MonthlyUsage	The total monthly usage of the customer (in minutes or GB)	Numerical	Integer, between 10 and 300
SupportCalls	The number of customer service calls made by the customer	Numerical	Integer, between 0 and 10

Column Name	Description	Type	Range
Churn	Whether the customer has churned or not (1 = Churn, 0 = No Churn)	Binary (Target)	0 or 1

---

4. Tentative Steps

The following steps outline the process for solving the churn prediction problem:

1. **Data Loading and Preprocessing**
    - Load the dataset and check for missing values.
    - Handle missing data either by imputing or removing it.
    - Convert categorical variables (ServicePlan, PaymentMethod) into numeric representations using encoding techniques (e.g., one-hot encoding).
    - Scale numerical features using standardization techniques (e.g., StandardScaler or MinMaxScaler).
2. **Exploratory Data Analysis (EDA)**
    - Analyze the relationships between features and the target variable (Churn).
    - Visualize correlations and identify any multicollinearity among features.
    - Create visualizations such as histograms, box plots, and scatter plots to understand feature distributions and relationships.
3. **Data Splitting**
    - Split the dataset into training and test sets, typically using an 80%-20% ratio.
4. **Model Building**
    - Apply multiple machine learning algorithms to build predictive models for customer churn:
        - **K-Nearest Neighbors (KNN)**
        - **Naïve Bayes**
        - **Decision Trees**
        - **Random Forest**
        - **AdaBoost**
        - **XGBoost**
        - **Support Vector Machine (SVM)**

5. **Model Evaluation**
    - Evaluate the performance of each model using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC curve.
    - Visualize confusion matrices and classification reports to assess model performance.
6. **Model Tuning (Hyperparameter Optimization)**
    - Tune hyperparameters of models like KNN, Decision Trees, Random Forest, XGBoost, and SVM to improve performance.
    - Use techniques such as Grid Search for hyperparameter tuning.
7. **Model Refinement and Selection**
    - After evaluation, choose the best-performing model based on its metrics and overall accuracy.
    - Fine-tune the selected model to further improve its performance if needed.
8. **Conclusion**
    - Summarize the final model's performance, the importance of features, and its potential real-world applications.
    - Suggest further improvements or alternative approaches if necessary.

---

5. Data Preprocessing

Before applying machine learning models, the dataset needs to be preprocessed:

- **Handle Missing Values**: Use techniques like **KNN Imputation** to handle missing values in features such as MonthlyUsage.
- **Encode Categorical Features**: Convert categorical variables (ServicePlan, PaymentMethod) into numerical representations using **OneHotEncoding or LabelEncoding**.
- **Scale Numerical Features**: Apply **StandardScaler** or **MinMaxScaler** to standardize the numerical features (Age, Tenure, SupportCalls, etc.) for better model performance.

---

6. Exploratory Data Analysis (EDA)

EDA is performed to understand the relationships between features and the target variable (Churn).

- **Visualizing Correlations**: Use heatmaps to check correlations between the features and the target variable.
- **Feature-Target Relationship**: Create scatter plots, bar plots, and box plots to understand how features such as Age, MonthlyUsage, and SupportCalls influence churn.

---

7. Model Building

1. **Data Splitting**: Split the data into training and testing sets (e.g., 80% for training, 20% for testing).
2. **Model Training**: Apply multiple classification algorithms to train models on the training dataset:
    - **KNN**: Train a KNN classifier to predict churn based on the nearest neighbors.
    - **Naïve Bayes**: Use Naïve Bayes to classify customers based on the probability of churn.
    - **Decision Trees**: Use decision trees to build an interpretable model based on entropy and information gain.
    - **Ensemble Models**: Train **Random Forest**, **AdaBoost**, and **XGBoost** for better accuracy and robustness.
    - **Support Vector Machine (SVM)**: Train a Support Vector Machine model to classify customers into churned or non-churned classes.
3. **Model Evaluation**: Evaluate each model's performance using classification metrics (accuracy, precision, recall, F1-score, etc.).

---

8. Model Evaluation

Evaluate model performance using metrics such as:

- **Accuracy**: Proportion of correctly classified samples.
- **Precision**: Accuracy of positive predictions (useful for churn prediction).
- **Recall**: Ability to identify actual churn cases.
- **F1-Score**: Balance between precision and recall.
- **AUC-ROC**: Evaluates the model's ability to distinguish between churn and non-churn.

Additionally, visualize the **confusion matrix** to better understand model performance.

---

9. Interpretation of Results

- **Coefficients and Feature Importance**: Examine feature importance in decision tree-based models (e.g., Random Forest, XGBoost). Features like MonthlyUsage or SupportCalls may play a key role in predicting churn.
- **SVM Hyperplane**: For SVM, we will analyze the position of the hyperplane and support vectors. The margin between the classes (churn vs. no-churn) and the decision boundary will provide insights into how well the SVM is separating the classes.

---

10. Model Improvement (if necessary)

If the models are not performing well:

- **Hyperparameter Tuning**: Use **GridSearchCV** to fine-tune hyperparameters like the number of estimators in **Random Forest** or **XGBoost**.
- **SVM Optimization**: Adjust SVM hyperparameters, such as the kernel type (linear, radial basis function, etc.), to improve model performance.

---

11. Conclusion

In this case study, various machine learning models to be applied to predict customer churn for a telecom company. After preprocessing the data and evaluating multiple models, select the best-performing model based on the classification metrics. By predicting which customers are at risk of churning, the company can take targeted actions to retain high-risk customers and improve their overall business performance.