Assignment: AI System Design

1. Problem Definition

AI Problem: Predicting Customer Churn for a Telecommunications Company.

This is a classification problem where the goal is to identify customers who are highly likely to cancel their subscriptions shortly. By predicting churn, the company can intervene with targeted actions to retain valuable customers.

Objectives:

Reduce Customer Attrition: Proactively identify at-risk customers and offer them incentives, support, or better plans to prevent them from leaving.

Increase Customer Lifetime Value (CLV): By retaining customers for longer periods, the company maximizes the total revenue generated from each customer.

Optimize Retention Spending: Focus marketing and retention budgets on customers with a high probability of churning, avoiding costly blanket offers to the entire customer base.

Stakeholders:

Marketing & Retention Teams: These teams are the primary users of the model's output. They will use the list of at-risk customers to launch targeted email campaigns, special promotions, and personalized outreach calls.

Executive Leadership (CEO, CFO): This group monitors the company's overall health. They use the churn predictions and outcomes to gauge business stability, forecast revenue, and report to investors.

Key Performance Indicator (KPI):

Quarterly Churn Rate Reduction: The primary measure of success will be the percentage decrease in the customer churn rate each quarter. The goal is to achieve and maintain a **10% reduction** in the churn rate within six months of deploying the model.

2. Data Collection & Preprocessing

Data Sources:

Customer Relationship Management (CRM) Database: This system provides static customer data, including demographics (age, location), account information (contract type: month-to-month, one-year, two-year), services subscribed to (e.g., phone, internet, streaming), and billing history (monthly charges, payment method).

Customer Service Interaction Logs: This source contains records of every interaction a customer has had with support, including the number of support tickets raised, the reasons for contact (e.g., technical issues, billing disputes), and the time to resolution.

Potential Bias:

Historical Bias: The data may reflect past business practices that are no longer relevant or were inherently biased. For example, if the company previously offered a problematic service plan that caused a high number of customers to churn, the model might learn to heavily penalize features associated with that plan. This could lead to inaccurate predictions for new customers who are on different, more stable plans.

Preprocessing Steps:

Handling Missing Data: Customer records may be incomplete (e.g., a missing 'age' value). For numerical columns like age or tenure, we will use **mean imputation**, filling in the missing values with the average value from the entire dataset. For categorical data, we could use the mode (the most frequent category).

Categorical Feature Encoding: The AI model requires numerical input. We will convert categorical features like Contract Type ('Month-to-Month', 'One-Year', 'Two-Year') into numbers using **One-Hot Encoding**. This creates new binary (0/1) columns for each category, preventing the model from assuming a false order between them.

Feature Scaling (Normalization): Features like Monthly Charges (e.g., $20 - $120) and Number of Support Tickets (e.g., 0 - 15) exist on vastly different scales. This can cause the model to incorrectly assign more importance to features with larger numerical values. We will use **Min-Max Scaling** to transform all numerical features to a common range of 0 to 1, ensuring they contribute equally to the model's predictions.

3. Model Development

Choice of Model & Justification:

Model: Gradient Boosting Machine (like XGBoost or LightGBM)

Justification: Gradient Boosting is a powerful and widely used algorithm for tabular data like ours.

High Accuracy: It is known for its high predictive accuracy and often wins data science competitions.

Handles Complex Relationships: It can capture non-linear relationships between features (e.g., the impact of high data usage on churn might be different for customers on unlimited plans versus those on capped plans).

Feature Importance: Like Random Forest, it can provide a clear ranking of which features (e.g., Contract Type, Tenure) are most predictive of churn, which provides actionable insights for the business.

Data Splitting:

The dataset will be partitioned into three distinct sets to ensure a robust and unbiased evaluation:

Training Set (70%): Used to train the Gradient Boosting model on the historical data.

Validation Set (15%): Used to tune the model's hyperparameters and select the best-performing version of the model without touching the test set.

Test Set (15%): A completely unseen dataset that is used only once at the end to provide a final, unbiased assessment of the model's performance on new data.

Hyperparameters to Tune:

learning_rate: This parameter controls the step size at each iteration while moving toward a minimum loss function.

Why Tune? A learning rate that is too high can cause the model to miss the optimal solution, while one that is too low can make the training process excessively slow. We would tune this to find a balance between speed and accuracy.**n_estimators:** This is the total number of sequential trees to be built.

Why Tune? More trees can increase accuracy but also increase the risk of overfitting the training data and add to the computational expense. We would use the validation set to find the optimal number of trees before performance starts to degrade.

4. Evaluation & Deployment

Evaluation Metrics:

Recall (Sensitivity): This metric measures the model's ability to find all the actual positive cases. It is calculated as (True Positives) / (True Positives + False Negatives).

Relevance: For churn prediction, Recall is critical. A "False Negative" is a customer whom the model failed to flag, but who then churned. This represents a lost opportunity for retention. We want to maximize Recall to catch as many at-risk customers as possible, even if it means we incorrectly flag a few safe customers (lower precision). The cost of losing a customer is much higher than the cost of making an unnecessary retention offer.

F1-Score: This is the harmonic mean of Precision and Recall (2 * (Precision * Recall) / (Precision + Recall)).

Relevance: Since focusing only on Recall might lead to a model that flags too many customers, the F1-Score provides a balanced measure. It is useful when you need to find a compromise between identifying all actual churners (Recall) and not wasting resources on customers who are not at risk (Precision). It's a good overall metric for business problems with imbalanced classes like churn.

Concept Drift:

What it is: Concept drift occurs when the underlying relationships between input variables and the target variable change over time. In our case, customer behaviors leading to churn might change due to external factors like a new competitor launching an aggressive pricing campaign, a technology change (e.g., 5G rollout), or a shift in the economic climate. A model trained on old data will become less accurate as these new patterns emerge.

How to Monitor it: We would implement an automated monitoring system that periodically (e.g., weekly or monthly) calculates the model's **Recall and F1-Score** on the most recent batch of customer data. If these metrics show a consistent downward trend over several periods, an alert is triggered, signaling that concept drift is likely occurring and the model needs to be retrained on more recent data.

Technical Challenge during Deployment:

Data Pipeline Latency: For the model to be truly useful, it needs fresh data. A significant technical challenge is building a reliable and low-latency data pipeline that can collect and preprocess data from various sources (CRM, usage logs, support tickets) in near real-time. If the data pipeline is too slow, the model will be making predictions based on outdated information (e.g., predicting churn for a customer who has already resolved their issue with support), making its output irrelevant or incorrect. Ensuring this pipeline is robust, scalable, and fast is a major engineering effort.