Part 1: Short Answer Questions (30 points)

1. Problem Definition (6 points)

Problem: Predicting customer churn for a telecommunications company to identify customers likely to cancel their subscription within the next 3 months.

3 Objectives:

1. **Reduce Revenue Loss:** Minimize financial impact by identifying at-risk customers before they churn, enabling proactive retention strategies
2. **Improve Customer Retention Rate:** Increase overall customer lifetime value by reducing churn rate from current 15% to below 10% within 12 months
3. **Optimize Marketing Spend:** Target retention campaigns more effectively by focusing resources on customers with highest churn probability and retention potential

2 Stakeholders:

1. **Marketing Team:** Uses predictions to design targeted retention campaigns, personalized offers, and customer engagement strategies
2. **Customer Success Managers:** Leverage churn predictions to prioritize outreach efforts and provide proactive support to at-risk customers

1 Key Performance Indicator (KPI):

Customer Retention Rate Improvement: Measure the percentage increase in customers retained after implementing the churn prediction model, specifically tracking the reduction in monthly churn rate and the ROI of retention campaigns initiated based on model predictions.

---

2. Data Collection & Preprocessing (8 points)

2 Data Sources:

1. **Customer Relationship Management (CRM) System:** Contains customer demographics (age, location, contract type), service usage patterns (data consumption,

call duration, SMS usage), billing history, customer service interactions, and subscription details

2. **Network Performance Logs:** Technical data including call drop rates, network coverage quality, data speed metrics, service outages, and connection reliability statistics for each customer's location and usage patterns

1 Potential Bias:

Geographic Bias: Data may be skewed toward urban customers who have better network coverage and more service options. Rural customers might appear to have higher churn rates due to limited network quality rather than dissatisfaction with service offerings. This could lead the model to incorrectly prioritize network infrastructure issues over customer service improvements, potentially misallocating resources and creating unfair treatment of customers in different geographic regions.

3 Preprocessing Steps:

1. **Handling Missing Data:**
    - Use forward-fill method for time-series data like monthly usage patterns
    - Apply median imputation for numerical features like average monthly spend
    - Create binary flags for systematically missing data (e.g., customers without recorded support calls)
2. **Feature Engineering and Normalization:**
    - Create derived features like "usage trend" (increasing/decreasing over last 6 months) and "support ticket frequency"
    - Apply Min-Max scaling to numerical features (usage minutes, data consumption, billing amounts)
    - Encode categorical variables using one-hot encoding for contract types and target encoding for high-cardinality features like location
3. **Temporal Alignment and Aggregation:**
    - Aggregate daily usage data into monthly summaries to create consistent time windows
    - Create rolling averages and trend indicators for the past 3, 6, and 12 months
    - Ensure all features represent the same time period before the prediction window to prevent data leakage

---

3. Model Development (8 points)

Model Choice: Gradient Boosting (XGBoost)

Justification: XGBoost is ideal for churn prediction because:

- **Handles Mixed Data Types:** Effectively processes both numerical (usage patterns, billing amounts) and categorical features (contract type, location) without extensive preprocessing
- **Built-in Feature Importance:** Provides interpretable insights into which factors most influence churn, crucial for business stakeholders to understand and act upon
- **Robust to Imbalanced Data:** Typically only 10-20% of customers churn, and XGBoost handles class imbalance well with built-in techniques
- **High Performance:** Consistently performs well on tabular data with complex non-linear relationships between customer behaviors and churn probability

Data Splitting Strategy:

- **Training Set (60%):** Used for model learning and parameter estimation
- **Validation Set (20%):** Used for hyperparameter tuning, model selection, and preventing overfitting during development
- **Test Set (20%):** Reserved for final unbiased performance evaluation

Temporal Consideration: Ensure chronological splitting where training data comes from earlier time periods than validation/test data to simulate real-world deployment conditions and prevent data leakage.

2 Hyperparameters to Tune:

1. **Learning Rate (eta):**

    - **Range:** 0.01 to 0.3
    - **Why:** Controls how much each tree contributes to the final prediction. Lower values (0.01-0.05) prevent overfitting and improve generalization but require more trees. Higher values (0.1-0.3) speed up training but may cause the model to converge too quickly to suboptimal solutions.

2. **Max Depth:**

    - **Range:** 3 to 10
    - **Why:** Determines tree complexity and model's ability to capture interactions. Shallow trees (3-5) prevent overfitting and maintain interpretability, while deeper trees (6-10) can capture complex customer behavior patterns but risk overfitting, especially with limited data.

---

4. Evaluation & Deployment (8 points)

2 Evaluation Metrics:

1. **Precision (Positive Predictive Value):**
    - **Relevance:** Measures accuracy of churn predictions - what percentage of customers predicted to churn actually do churn
    - **Business Impact:** High precision reduces wasted marketing spend on false positives (customers incorrectly identified as likely to churn), ensuring retention campaigns target genuinely at-risk customers
2. **Recall (Sensitivity):**
    - **Relevance:** Measures the model's ability to identify actual churners - what percentage of customers who actually churn were correctly predicted
    - **Business Impact:** High recall ensures the company doesn't miss revenue-critical customers who are about to leave, maximizing the opportunity to retain valuable customers through intervention

Concept Drift Definition and Monitoring:

Concept Drift occurs when the statistical properties of the target variable (churn behavior) change over time, making the model less accurate as customer behavior patterns evolve due to market changes, new competitors, seasonal factors, or economic conditions.

Monitoring Strategy:

- **Performance Monitoring:** Track model accuracy, precision, and recall monthly using a rolling window of recent predictions vs. actual outcomes
- **Feature Distribution Monitoring:** Monitor key input features (usage patterns, payment behavior) for significant statistical changes using techniques like Kolmogorov-Smirnov tests
- **Automated Alerts:** Set up alerts when performance metrics drop below acceptable thresholds (e.g., precision <70%) or when feature distributions deviate significantly from training data
- **Retraining Schedule:** Implement quarterly model retraining with recent data, or trigger retraining when drift is detected

1 Technical Challenge During Deployment:

Real-time Inference Latency and Scalability:

Challenge Details: The model must process predictions for millions of customers efficiently while maintaining low latency for real-time applications like customer service representatives needing immediate churn risk assessments during support calls.

Specific Issues:

- **Data Pipeline Complexity:** Aggregating features from multiple systems (CRM, billing, network logs) in real-time creates bottlenecks
- **Model Complexity:** XGBoost with optimized hyperparameters may be too slow for real-time inference at scale
- **Infrastructure Scaling:** Peak usage times (month-end billing cycles) require dynamic scaling to handle increased prediction requests

Potential Solutions:

- Implement model serving with containerized microservices for horizontal scaling
- Use feature stores to pre-compute and cache frequently used features
- Consider model compression techniques or deploying lighter models for real-time use cases while maintaining batch processing for comprehensive analysis