TELECOM CUSTOMER CHURN PREDICTION

Complete Interview Preparation Guide

Project Highlights:

Dataset: 3,333 customers | 19 features

Best Model: XGBoost Classifier

Accuracy: 98% | Recall: 87%

Deployment: Streamlit Web App

1. PROJECT OVERVIEW

A machine learning classification project that predicts which telecom customers are likely to churn (cancel service), enabling proactive retention strategies.

Complete Workflow

Data Loading -> EDA -> Preprocessing (StandardScaler + SMOTE) -> Feature Selection (Top 10) -> Train-Test Split (80-20) -> Model Building (RFC & XGBoost) -> Evaluation -> Deployment (Streamlit)

2. BUSINESS PROBLEM

What is Customer Churn?

Customer churn refers to when a customer stops using a companys service (cancels their subscription). It is also called customer attrition.

Why is Churn Prediction Important?

- Churn Rate in Telecom: Typically 10-25% annually

- Re

Project Objective

Build a predictive model that identifies at-risk customers BEFORE they leave, enabling targeted retention campaigns.

3. DATASET DESCRIPTION

Total Rows: 3,333 customers | Total Columns: 19 features + 1 target | No missing values

Key Features

Feature	Type	Description
total_charge	Numerical	Total monthly bill ($)
customer_service_calls	Numerical	Number of calls to support
international_plan	Binary	Has international plan (0/1)
day_mins	Numerical	Daytime call minutes
voice_mail_plan	Binary	Has voicemail plan (0/1)
churn	Target	0 = Stayed, 1 = Churned

4. EXPLORATORY DATA ANALYSIS (EDA)

4.1 Target Variable Analysis

Class 0 (Stayed): 2,850 customers (85.5%) | Class 1 (Churned): 483 customers (14.5%)

KEY INSIGHT: Class imbalance detected! Requires SMOTE for handling.

4.2 Top Churn Predictors

Predictor	Impact	Churn Rate
Service Calls >= 4	HIGHEST RISK	45%+
International Plan = Yes	3x higher	28% vs 11%
Total Charge > $65	Bill shock	Higher
No Voice Mail Plan	50% more	16% vs 8%

4.3 Key Correlation Findings

- international_plan with churn: +0.26 (positive = increases churn)
- cus

5. DATA PREPROCESSING

5.1 Standard Scaling

StandardScaler transforms features to have mean=0 and std=1.

Formula: z = (x - mean) / std

Why: Features have different scales (day_mins: 0-400 vs international_charge: 0-5)

5.2 SMOTE for Class Imbalance

SMOTE = Synthetic Minority Oversampling Technique

Before SMOTE: Class 0 = 2,850 | Class 1 = 483

After SMOTE: Class 0 = 2,850 | Class 1 = 2,850 (Balanced!)

How: Creates synthetic minority samples by interpolating between existing samples

6. FEATURE SELECTION

Random Forest Feature Importance

Used Random Forest to rank features by importance (Gini impurity reduction)

Rank	Feature	Importance
1	total_charge	21.3%
2	customer_service_calls	12.5%
3	day_mins	8.9%
4	day_charge	8.8%
5	international_plan	7.9%

7. MODEL BUILDING

7.1 Train-Test Split

80% Training (2,666 samples) | 20% Testing (667 samples)

7.2 Model Comparison

Metric	Random Forest	XGBoost	Winner
Accuracy	97%	98%	XGBoost
Recall (Churners)	81%	87%	XGBoost
F1-Score	0.89	0.93	XGBoost
False Negatives	20	13	XGBoost

7.3 Why XGBoost?

- Gradient boosting - sequential error correction

- Bu

8. MODEL EVALUATION

XGBoost Confusion Matrix

True Negatives: 566 | False Positives: 0

False Negatives: 13 | True Positives: 88

Classification Metrics

Metric	Value	Meaning
Accuracy	98%	Overall correct predictions
Precision	100%	When we predict churn, we are right
Recall	87%	We catch 87% of actual churners
F1-Score	0.93	Balanced metric

Why Recall Matters More

False Negative (missing a churner) = $780/year lost revenue

False Positive (false alarm) = Small cost (unnecessary discount)

9. CONCLUSIONS & RECOMMENDATIONS

Key Findings

- 14.5% churn rate - typical for telecom
- Se

Business Recommendations

- Flag customers after 2nd service call for monitoring
- As

Business Impact

With 100,000 customers, this model can save $5+ million annually through proactive retention.

10. INTERVIEW QUESTIONS & ANSWERS (PART 1)

Q1: Tell me about your project

I developed a Customer Churn Prediction system for telecom. Using 3,333 customers with 19 features, I built an XGBoost classifier achieving 98% accuracy and 87% recall.

Q2: What was the business problem?

Telecom companies face 10-25% annual churn. Each lost customer costs $60-80/month. My model identifies at-risk customers early, saving millions in revenue.

Q3: What was your methodology?

EDA -> Preprocessing (Scaling + SMOTE) -> Feature Selection (Random Forest) -> Model Building (RFC vs XGBoost) -> Evaluation -> Streamlit Deployment.

Q4: What patterns did you find in EDA?

1) Service calls 4+ = 45%+ churn, 2) International plan = 3x churn, 3) Voicemail plan reduces churn by 50%.

Q5: How did you handle outliers?

Kept them! They represent real customers (heavy users/dissatisfied) who are exactly the churners we want to predict. XGBoost handles trees robustly.

Q6: What did correlation analysis show?

High correlation (0.99) between minutes and charges. Service calls and international plan showed positive correlation with churn.

11. INTERVIEW Q&A (PART 2)

Q7: Why StandardScaler?

Normalizes features to Mean=0, Std=1. Prevents features with large ranges (day_mins) from dominating those with small ranges (charge).

Q8: How does SMOTE work?

Creates synthetic minority samples by interpolating between existing ones (k-nearest neighbors) rather than just duplicating records.

Q9: Random Forest vs XGBoost?

RF builds trees in parallel (bagging). XGBoost builds them sequentially (boosting) to correct previous errors, yielding 98% vs 97% accuracy.

Q10: Explain Precision vs Recall

Precision (100%): No false alarms. Recall (87%): Caught 87% of actual churners. For churn, Recall is more critical to avoid losing revenue.

Q11: Why is 98% accuracy misleading?

In imbalanced data (85% stay), a dummy model predicting "no churn" gets 85% accuracy but fails the business objective. We use Recall and F1-Score.

Q12: Business impact calculation?

With 100K users and 15% churn, model identifies 13,050. With 50% retention success, saves $5.09M annually at $65/month revenue.

11. INTERVIEW Q&A (PART 3)

Q13: How does Gradient Boosting work?

Builds trees sequentially where each tree predicts the residuals (errors) of the previous trees, progressively reducing the error.

Q14: What hyperparameters would you tune?

n_estimators, max_depth, learning_rate, and scale_pos_weight (for imbalance) using GridSearchCV or RandomizedSearchCV.

Q15: Explain the Confusion Matrix

TN: 566 (Correct stays), TP: 88 (Correct churns), FP: 0 (No false alarms), FN: 13 (Missed churners).

Q18: What is F1-Score?

Harmonic mean of precision and recall. It penalizes extreme values and provides a balanced measure for imbalanced datasets.

Q19: Explain SMOTE technically

Identifies k-nearest neighbors for minority samples and creates new points along the lines connecting them to increase diversity.

Q21: How would you deploy in production?

Containerize with Docker, deploy on cloud (AWS/GCP), implement model monitoring for drift, and use A/B testing.

11. INTERVIEW Q&A (PART 4)

Q23: Business recommendations?

1) Intervene at 3rd service call, 2) Review intl plan pricing, 3) Use voicemail as a "sticky" feature to reduce churn.

Q24: What would you do differently?

Use k-fold cross-validation, more automated hyperparameter tuning, and add model interpretability tools like SHAP.

Q26: Cost of False Negatives?

Missing a churner costs ~$780/year in lost revenue. This is why we prioritize Recall over Precision.

Q27: How to monitor drift?

Track prediction distribution and feature distributions over time. Retrain model if performance drops significantly.

Q30: Why not Deep Learning?

For tabular data of this size, Gradient Boosting (XGBoost) typically outperforms Deep Learning and is much more interpretable.

12. MACHINE LEARNING FUNDAMENTALS

Q31: Bias-Variance Tradeoff

Bias is error from wrong assumptions (underfitting). Variance is error from sensitivity to noise (overfitting). Goal is to find the balance.

Q32: Overfitting vs Underfitting

Overfitting: Low train error, High test error (too complex). Underfitting: High train error, High test error (too simple).

Q33: L1 vs L2 Regularization

L1 (Lasso) adds absolute penalty and can do feature selection. L2 (Ridge) adds squared penalty. XGBoost uses both to control complexity.

13. CODING & SQL CHALLENGES

Python: Calculate Churn Rate

```
def calculate_churn(statuses):
    return (sum(statuses) / len(statuses)) * 100
# Example: [0, 1, 0, 1] -> 50%
```

SQL: High Value State Analysis

```
SELECT state, COUNT(id), AVG(charge)
FROM users GROUP BY state
HAVING COUNT(id) > 50 ORDER BY AVG(charge) DESC;
```

14. QUESTIONS TO ASK THE INTERVIEWER

- How does the company handle retention for customers flagged by ML models?
- Wh

QUICK REFERENCE CARD

Item	Value
Best Model	XGBoost Classifier
Accuracy / Recall	98% / 87%
Top Predictor	Service Calls (>=4)
Preprocessing	StandardScaler + SMOTE
Annual Savings	$5.09 Million

GOOD LUCK WITH YOUR INTERVIEW!