1. How would you handle imbalanced data if churned customers are fewer than active ones?
Ans-:
Resampling Techniques:
Oversampling: Use techniques like SMOTE (Synthetic Minority Oversampling Technique) to generate synthetic samples for the minority class.
Undersampling: Reduce the number of samples in the majority class to balance the dataset.

Feature Engineering:
Create meaningful features to help the model better distinguish churned from active customers.

2. What features are the most important predictors of churn?
Ans-:
Recency
Income
NumWebVisitsMonth
MntWines, MntMeatProducts, MntGoldProds
NumDealsPurchases, NumWebPurchases, NumStorePurchases
AcceptedCmp1, AcceptedCmp2, AcceptedCmp3
Complain
Education
Marital_Status

3.How would you explain the model's predictions to a non-technical business team?
Ans-:
The model predicts whether a customer is likely to stop engaging with our services based on their spending habits, engagement levels, and responses to marketing campaigns.Highlight the most important predictors (e.g., Recency, NumWebVisitsMonth, Income, and spending in specific categories like MntWines or MntMeatProducts).
Example: "If a customer hasn't made a purchase in a long time (Recency is high) or spends less on products they used to buy (MntWines), the model flags them as likely to churn.Use simple bar charts or heatmaps to show feature importance.
Example: "Here’s a chart showing that recent activity (Recency) and total spending (Income) are the top factors influencing the prediction".

4.What steps would you take to deploy this model into production?
Ans-:
1. Data Preprocessing-:
Missing Values Handling: Manage missing data in fields like Income if applicable.
Encoding: Convert categorical fields like Education and Marital_Status into numerical formats.
Scaling: Normalize numerical features like Income, Recency, and spending attributes (MntWines, etc.).

Date Features: Convert Dt_Customer into derived features (e.g., customer tenure).
2. Model Training and Validation-:
Train the model using algorithms like Random Forest, XGBoost, or Logistic Regression with balanced class handling.
Validate the model using cross-validation and metrics relevant to churn, such as AUC-ROC, precision, and recall.
3. API Development-:
Wrap the model in a RESTful API using frameworks like Flask, FastAPI, or Django.
Ensure the API takes input features (Recency, Income, etc.) in the correct format and returns the churn probability.
4. Integration with Production Systems
Integrate the API with the CRM system to make predictions in real time or batch mode.
Store predictions alongside customer data to enable action (e.g., retention strategies).
5. Automation
Feature Engineering Automation: Automate feature creation and transformation for incoming data.
Deployment Pipelines: Use tools like Docker and Kubernetes to containerize and deploy the API for scalability.
6. Monitoring and Retraining
Monitor Predictions: Track model accuracy, drift in features (e.g., changes in Income distribution), and latency.
Retraining: Regularly retrain the model with updated data to account for changes in customer behavior.