**Approach Document for Personal Loan Prediction Project**

**1. Problem Statement**

AllLife Bank aims to expand its asset customer base by converting liability customers (depositors) into personal loan customers. While the bank’s previous marketing campaigns achieved a 9% conversion rate, the goal is to optimize future campaigns by identifying potential customers with a higher likelihood of purchasing personal loans.

As a data scientist, the objective is to build a robust predictive model to:

* Predict whether a liability customer will purchase a personal loan.
* Understand which customer attributes significantly influence loan purchases.
* Recommend customer segments to target for future campaigns.

**2. Data Understanding**

**Data Overview**

The dataset provided contains the following columns:

* **ID**: Unique identifier for each customer.
* **Age**: Customer’s age in years.
* **Experience**: Years of professional experience.
* **Income**: Annual income in thousand dollars.
* **ZIP Code**: Home address ZIP code.
* **Family**: Family size of the customer.
* **CCAvg**: Average monthly spending on credit cards.
* **Education**: Education level (1 = Undergrad, 2 = Graduate, 3 = Advanced/Professional).
* **Mortgage**: Value of house mortgage (in thousand dollars).
* **Personal\_Loan**: Target variable (0 = No, 1 = Yes).
* **Securities\_Account**: Whether the customer has a securities account (0 = No, 1 = Yes).
* **CD\_Account**: Whether the customer has a certificate of deposit account (0 = No, 1 = Yes).
* **Online**: Whether the customer uses internet banking (0 = No, 1 = Yes).
* **CreditCard**: Whether the customer uses a credit card issued by other banks (0 = No, 1 = Yes).

**3. Exploratory Data Analysis (EDA)**

**Key Observations**

* **Demographic Trends**:
  + Higher income levels correlate with a higher likelihood of accepting personal loans.
  + Customers with advanced education levels are more inclined to purchase loans.
* **Financial Behavior**:
  + Customers with high average credit card spending (CCAvg) and mortgage values are more likely to take loans.
* **Banking Behavior**:
  + Customers using online banking or holding CD accounts are more likely to accept personal loans.
* **Target Variable**:
  + Class imbalance observed: Only ~9% of customers accepted personal loans, requiring careful handling during modeling.

**4. Data Preprocessing and Feature Engineering**

**Steps Taken**

1. **Outlier Treatment**:
   * Applied IQR-based filtering to remove outliers in the majority class.
2. **Feature Engineering**:
   * Created new features:
     + Income\_per\_Family: Ratio of income to family size.
     + Mortgage\_to\_Income: Ratio of mortgage value to income.
   * Binned income into categories (Low, Medium, High, Very High) for better segmentation.
3. **Encoding and Scaling**:
   * Applied OneHotEncoder for categorical variables.
   * Scaled numerical features using StandardScaler.
4. **Train-Test Split**:
   * Split the dataset into training (70%) and testing (30%) subsets with stratification to address class imbalance.

**5. Model Building and Evaluation**

**Initial Model: Decision Tree Classifier**

* Achieved high training accuracy but overfit the data, evident from the large tree size and slight performance drop on test data.

**Pruning for Improved Performance**

* Applied tree pruning to:
  + Enhance interpretability.
  + Prevent overfitting.
  + Achieve balanced generalization.

**Final Model Results**

**Before Pruning:**

* Train Accuracy: 100% | Test Accuracy: 97.48%
* Precision: 90.85% | Recall: 89.58% | F1 Score: 90.21%

**After Pruning:**

* Train Accuracy: 95.40% | Test Accuracy: 94.80%
* Precision: 65.14% | Recall: 98.61% | F1 Score: 78.45%

Pruning resulted in a more realistic and interpretable model while maintaining strong performance metrics. This pruned decision tree was selected as the **final model**.

**6. Deployment Strategy**

To operationalize the solution, we will create an interactive and user-friendly web application using the following tools:

**Tools and Technologies**

* **Backend**:
  + Flask: Lightweight Python web framework for serving the model.
* **Frontend**:
  + HTML, CSS, and JavaScript: To design an intuitive interface for users.
  + Interactive forms for customer data input.
* **Deployment**:
  + Render or Heroku: For deploying the web application to the cloud.

**Deployment Workflow**

1. **Model Serialization**:
   * Use joblib or pickle to save the trained model.
2. **API Development**:
   * Develop a Flask API endpoint that:
     + Accepts customer data.
     + Processes the input using the preprocessor pipeline.
     + Returns loan eligibility predictions.
3. **Frontend Interface**:
   * Design a form where bank staff can input customer details.
   * Display the prediction result along with actionable insights.
4. **Cloud Deployment**:
   * Deploy the Flask application using Render or Heroku for easy accessibility.

**7. Next Steps**

1. **Develop Flask Backend**:
   * Build API endpoints for prediction and result handling.
2. **Design Frontend Interface**:
   * Create responsive and visually appealing forms.
3. **Integrate Model and Interface**:
   * Connect the frontend with Flask endpoints.
4. **Cloud Deployment**:
   * Test and deploy the application on a cloud platform.
5. **User Feedback**:
   * Gather feedback from stakeholders to improve the interface and prediction system.

**8. Actionable Insights and Recommendations**

1. **Customer Segmentation**:
   * Focus on high-income families, customers with advanced education levels, and those with high credit card spending.
2. **Marketing Campaigns**:
   * Personalize offers for customers actively using online banking and CD accounts.
   * Highlight benefits of loan offers to advanced education-level customers.
3. **Future Enhancements**:
   * Explore advanced ensemble models for improved precision.
   * Incorporate customer feedback for better campaign strategies.