**Project Summary**

**Customer Segmentation in Banking**

Group Name: Finance Bank

Feyzanur Akdeniz 2103943

Tan Alaeddinoğlu 2004155

Kaan Ateş 2004003

**Problem Definition**

Banks aim to enhance customer experience and operational efficiency by understanding their customers better. Customer segmentation helps categorize customers into distinct groups based on their behaviors and characteristics. This segmentation allows for tailored marketing strategies, personalized services, and efficient allocation of resources.  
The objective of this project is to implement machine learning and deep learning techniques to analyze the given customer data, extract meaningful patterns, and create well-defined customer segments. These insights can drive strategic decision-making for banks, ultimately improving customer satisfaction and business outcomes.

**Used Tools:**

Libraries: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, SciPy.

Pyhton

Google Collab

**Dataset Overview**

The dataset for this project is sourced from Kaggle and contains detailed customer information to facilitate segmentation. It consists of two files: Train.csv and Test.csv.

Key attributes in the dataset include:

* **CustomerID**: A unique identifier for each customer.
* **Gender**: Gender of the customer (Male/Female).
* **Age**: Age of the customer.
* **Graduated**: Is the customer a graduate?
* **Profession:** Profession of the customer.
* **Ever Married:** Marital status of the customer.
* **Work\_Experience:** Work Experience in years.
* **Spending Score**: A score assigned by the bank based on customer behavior and spending patterns.
* **Family Size:** Number of family members for the customer (including the customer).
* **Segmentation:** (target) Customer Segment of the customer.

Segmentatin is already known the main purpose is after the all process we will cross check the predictions.

**Preprocessing and Explotary Data Analysis(EDA)**

The following steps were implemented:

* **Splitting Features and Target Variable**
  + The ID column was dropped as it does not contribute to the prediction task.
  + Features were split into numerical and categorical columns for targeted preprocessing.
* **Handling Missing Values**
* Numerical columns were imputed with the mean value using SimpleImputer(strategy='mean').
* Categorical columns were imputed with the most frequent value using SimpleImputer(strategy='most\_frequent').
* **Encoding Categorical Features**
* Label encoding was applied to transform categorical values into numerical representations, making them compatible with machine learning algorithms.
* **Combining Features**
* The preprocessed numerical and categorical features were concatenated into final feature matrices for both training and test datasets.
* **Target Variable Transformation**
* The target variable was encoded using LabelEncoder.

For exploratoy part these step were implemented:

* **Numerical Features**:
* Visualize the distribution of numerical columns (e.g., Age, Annual Income, and Spending Score) using histograms
* **Correlations Between Numerical Features**:
* A correlation matrix with a heatmap highlights relationships between numerical features.
* **Scatter Plots:**
  + Use scatter plots to explore relationships between key features like Annual Income and Spending Score, with potential segmentation visualized.
* **Calculate Class Weights:**
  + Class weights were computed using compute\_class\_weight to address any class imbalance in the dataset.

**Model Evaluation and Validation**

* **KNN Model Training:**
* The KNN algorithm is implemented using KNeighborsClassifier from sklearn.
* The number of neighbors, n\_neighbors, is varied to find the optimal number for the best classification performance.
* **Hyperparameter Tuning (Neighbors Selection):**
* We test multiple values of n\_neighbors ranging from 1 to 15.
* We use elbow method for the best k variable
* For each value of n\_neighbors, the KNN classifier is trained and tested on both the training and test sets.
* **Various evaluation metrics are calculated to assess the model’s performance:**
  + Accuracy(accuracy\_train, accuracy\_test): The ratio of correctly predicted instances to the total instances.
  + Precision(precision\_train, precision\_test):The ratio of correctly predicted positive observations to the total predicted positives.
  + Recall(recall\_train, recall\_test):The ratio of correctly predicted positive observations to the all observations in actual class.
  + F1-Score(f1\_train, f1\_test):The weighted average of Precision and Recall.

**Results and Post-Training Analysis**

* **All metric ( including accuracy) are in range of %40-50.**
* **Corelation Matrix Analysis:**
  + For the accuracy and consistency we check the matrix and scores
* **Creating .csv file**:
  + To display easily and understandable put the predictions and target values in a .csv file.