ANN Bank Churn Classification

This project is a **Bank Customer Churn Prediction System** built using an **Artificial Neural Network (ANN)**. It predicts whether a customer will leave the bank or not, based on various input features.  
  
Developed and deployed by **Omkar Gadade (PGA28)**, this project uses a combination of **VS Code, TensorFlow/Keras, and Streamlit** to create and deploy a web application for classification.

# Live Demo

<https://ann-bank-churn-classification-5vedds2uvjspbvvrn2zekn.streamlit.app/>

# Git Hub Repo

<https://github.com/Omkar-Gadade/ANN-Bank-Churn-Classification>

# Tools & Technologies Used

- **VS Code** – Development Environment  
- **Streamlit** – Web Application Framework  
- **Streamlit Cloud** – App Deployment  
- **Python Libraries**:  
 - pandas, numpy – Data Processing  
 - tensorflow, keras – Model Building  
 - scikit-learn – Preprocessing & Metrics  
 - ipykernel, tensorboard – Notebook and Training Visualizations  
- **Excel** – Data Input/Analysis

# Dataset

The dataset used is a **.csv file** containing **10 input features** related to customer activity. These include demographic and behavioral data to classify the likelihood of customer churn.

# Data Preprocessing

**- Encoding**:  
 - One-Hot Encoding for categorical features  
 - Label Encoding where appropriate  
- **Standardization** of input features  
- **Data split** into training and test sets

# ANN Model Architecture

- **Input Layer**: 10 features  
- **Hidden Layers**:  
 - Hidden Layer 1 (HL1): ReLU activation  
 - Hidden Layer 2 (HL2): ReLU activation  
**- Output Layer**: Sigmoid activation (binary classification)  
- **Optimizer**: Adam  
- **Loss Function**: Binary Cross-Entropy  
- **Metrics**: Accuracy

# Model Training

Model trained using **TensorFlow/Keras** and visualized via **TensorBoard** for:  
- Epoch-wise Accuracy  
- Epoch-wise Loss

# Model Files

- .**pkl** – Preprocessing pipeline  
- **.h5** – Trained ANN model

**experiments.ipynb** - The file is used to perform Feature Engineering and Preprocessing Tasks and saving the encodings and Model in '.pkl' and '.h5' files

**prediction.ipynb** - The file is used for just checking and experimenting the code for prediction before executing in the final app.py file.

**app.py** - The final main application .py file which is required to be run in vs code terminal

**requirements.txt** - pip install the requirements file to install all the dependencies and libraries (one can create a Virtual Environment)

# Streamlit Web App

A user-friendly interface was built using Streamlit to interact with the model in real time. Users can input feature values and get instant predictions on churn risk.  
  
App is hosted on Streamlit Cloud.  
  
GitHub Repo: https://github.com/Omkar-Gadade/ANN-Bank-Churn-Classification

# How to Run Locally

1. **Clone the Repository**:  
 git clone https://github.com/Omkar-Gadade/ANN-Bank-Churn-Classification.git  
 cd ANN-Bank-Churn-Classification  
  
2. **Create Virtual Environment**:  
 python -m venv venv  
 source venv/bin/activate (Windows: venv\Scripts\activate)  
  
3. **Install Requirements**:  
 pip install -r requirements.txt  
  
4. **Run the Streamlit App**:  
 streamlit run app.py

# Contact

**Author: Omkar Gadade**

omkarumeshgadade@gmail.com  
📍 Navi Mumbai, India