Simple Explanation of Finance Loan Agent Code Flow

Here's a simple explanation of how data moves and gets processed in your project:

Main Files and Their Functions

run\_agent.py - The file that starts the agent

• First gets the Google API key from the .env file

• Creates the loan agent from agent.py

• Takes input from the user (like "I need a loan of $25,000")

• Sends the input to the agent and displays the response

agent.py - The main agent file

• Connects to Google's Gemini AI model

• Links finance tools to the agent

• Sets up the AI to answer user questions

models/credit\_scoring.py - Credit scoring model

• Creates a deep learning model using TensorFlow

• Estimates the probability of loan default

• Learns from training data

tools/finance\_tools.py - Finance tools

• Analyzes loan applications

• Recommends interest rates

• Calculates monthly payments

tools/mongodb\_tools.py - MongoDB tools

• Connects to MongoDB

• Stores loan applications in the database

• Finds similar loans using vector search

train\_model.py - The file that trains the model

• Loads data from data/sampleloandata.csv

• Trains the credit scoring model

• Saves the model

• Creates text embeddings and stores them in MongoDB

Data Flow

Where does the data come from?

Sample Data: Already exists in the data/sampleloandata.csv file

• Contains loan application data (income, credit score, loan amount, etc.)

• Used for model training

User Input: When the user talks to the agent

• Example: "My annual income is $65,000, my credit score is 720, and I need a loan of $25,000"

Google API: Responses come from the Gemini AI model

• Your API key should be in the .env file

MongoDB: Previous loan applications and their embeddings are stored

• MongoDB connection string should be in the .env file

How is the data processed?

Training Process:

• train\_model.py loads sample data

• Splits data into features (X) and target (y)

• Trains the model

• Saves the model to models/creditscoringmodel.h5

• Creates text embeddings and stores them in MongoDB

Loan Application Process:

• User asks about a loan

• Agent understands the user's input

• Calls functions in finance\_tools.py

• Gets risk estimates from the credit scoring model

• Finds similar loans from MongoDB

• Calculates interest rate and monthly payments

• Responds to the user

Where is the data stored?

Model Storage:

• Trained model is saved in models/creditscoringmodel.h5

MongoDB Storage:

• Loan application data is stored in the loan\_applications collection

• Each application is stored with its text embedding

• A vector search index is created to find similar loans

Environment Variables:

• API keys and connection strings are stored in the .env file

How to Run the Project

First install dependencies:

pip install -r requirements.txt

Set up the .env file:

cp financeloanagent/.env.example financeloanagent/.env

Then add your Google API key and MongoDB connection string to the .env file

Train the model:

python financeloanagent/train\_model.py

Run the agent:

python financeloanagent/run\_agent.py

Talk to the agent, like:

My annual income is $65,000, my credit score is 720, and I need a $25,000 loan for 48 months

This is how your Finance Loan Agent processes data, stores it, and answers user questions.