BANK BOT AI CHATBOT HOR BANK FAQ'S

MILESTINE 1

ABSTRACT:

The project focuses on building an intelligent chatbot for banking systems using Natural Language Processing (NLP) techniques. The chatbot is trained to understand and classify customer queries such as checking balance, transferring money, blocking cards, and opening accounts. Using a labeled dataset, the model learns to identify user intent and extract key entities such as account type, card type, and transaction amount. This project demonstrates how AI can enhance banking automation, reduce manual effort, and improve customer support efficiency.

Introduction:

In modern banking, customers expect instant query handling. AI chatbots can automate responses to frequent customer requests, improving response time and efficiency.

This milestone focuses on developing the data preprocessing, training, and evaluation pipeline for a chatbot capable of understanding natural language queries.

Problem Statement:

Manual customer service in banks is time-consuming and prone to delays.

Customers face long waiting times for simple queries such as checking account balance or blocking a card.

The problem is to design an AI system that can:

Automatically identify the intent of the customer (e.g., transfer money, check balance).

Extract entities such as amount, card type, or account number.

Provide the correct automated response**.**

Objectives:

To collect and prepare a labeled dataset for chatbot training.

To use NLP (SpaCy + Scikit-learn) for intent classification.

To extract entities from customer queries.

To evaluate model performance using classification metrics.

Dataset Description:

File Name: traninig_and_responses.csv

Columns:

query: The customer's natural language question.

intent: The purpose of the query (e.g., check_balance, transfer_money).

entities: A dictionary of extracted information (e.g., {amount: '5000', receiver_name: 'John'}).

Example:

query intent entities

Show me the balance of my savings account 12345 check_balance{'account_type': 'savings', 'account_number': '12345'}

I want to transfer 5000 to John transfer_money {'amount': '5000', 'receiver_name': 'John'}

Methodology:

Step 1: Data Collection

A dataset containing 2000+ realistic banking queries is created in CSV format.

Step 2: Data Preprocessing

Load CSV data using pandas.
Convert entity strings into Python dictionaries using ast.literal_eval().
Split data into training and testing sets.

Step 3: Model Building

Use TfidfVectorizer to convert text queries into numerical vectors.
Train a Logistic Regression model using Scikit-learn pipeline.
Fit the model to classify intents.

Step 4: Model Evaluation

Evaluate using classification_report() to measure precision, recall, and F1-score.

Step 5: Output Generation

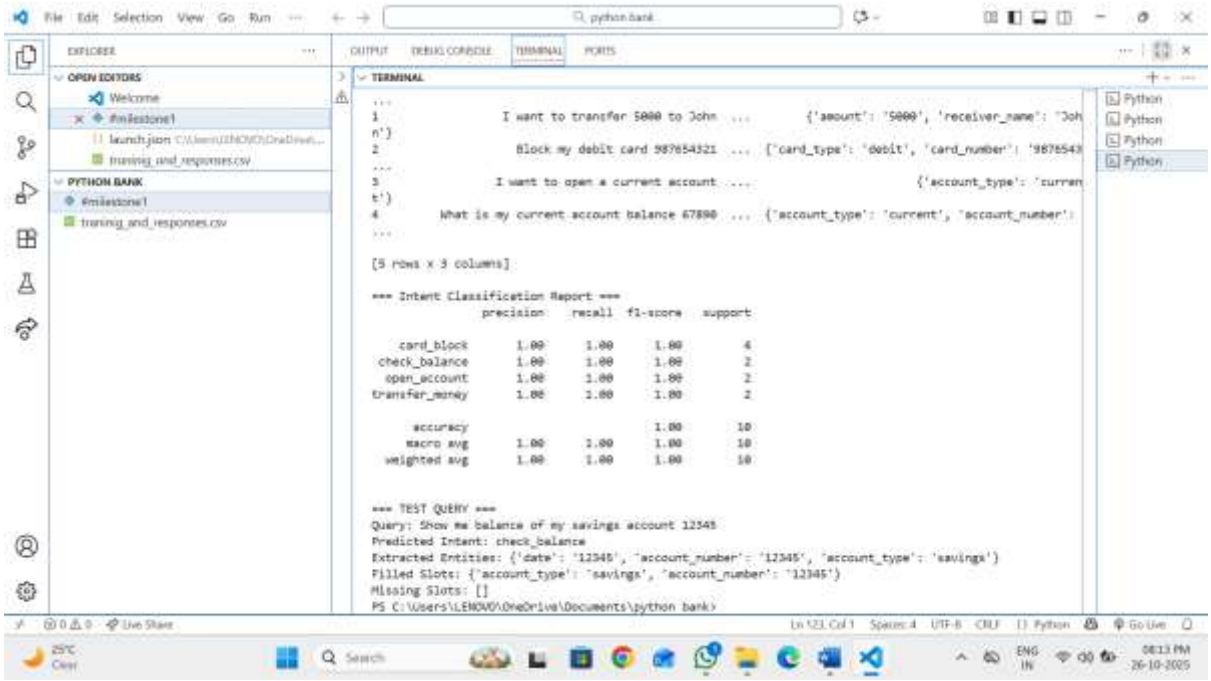
Test the chatbot with example queries.

Observe predicted intent and extracted entities.

Tools and Technologies Used:

Tool	Purpose
Python	Core programming language
Pandas	Data handling
Scikit-learn	Model building
SpaCy	NLP tokenization
TfidfVectorizer	Feature extraction
Logistic Regression	Classification model

Sample Output:



Conclusion:

The AI Bank Chatbot successfully classifies user intents and extracts entities from queries. It automates frequent banking tasks and forms the foundation for integrating with live APIs or web portals.

In the next milestone, integration with frontend and backend systems will enable a complete interactive banking assistant.