HCLTECH HACKATHON

– SUJĀN KUMAR

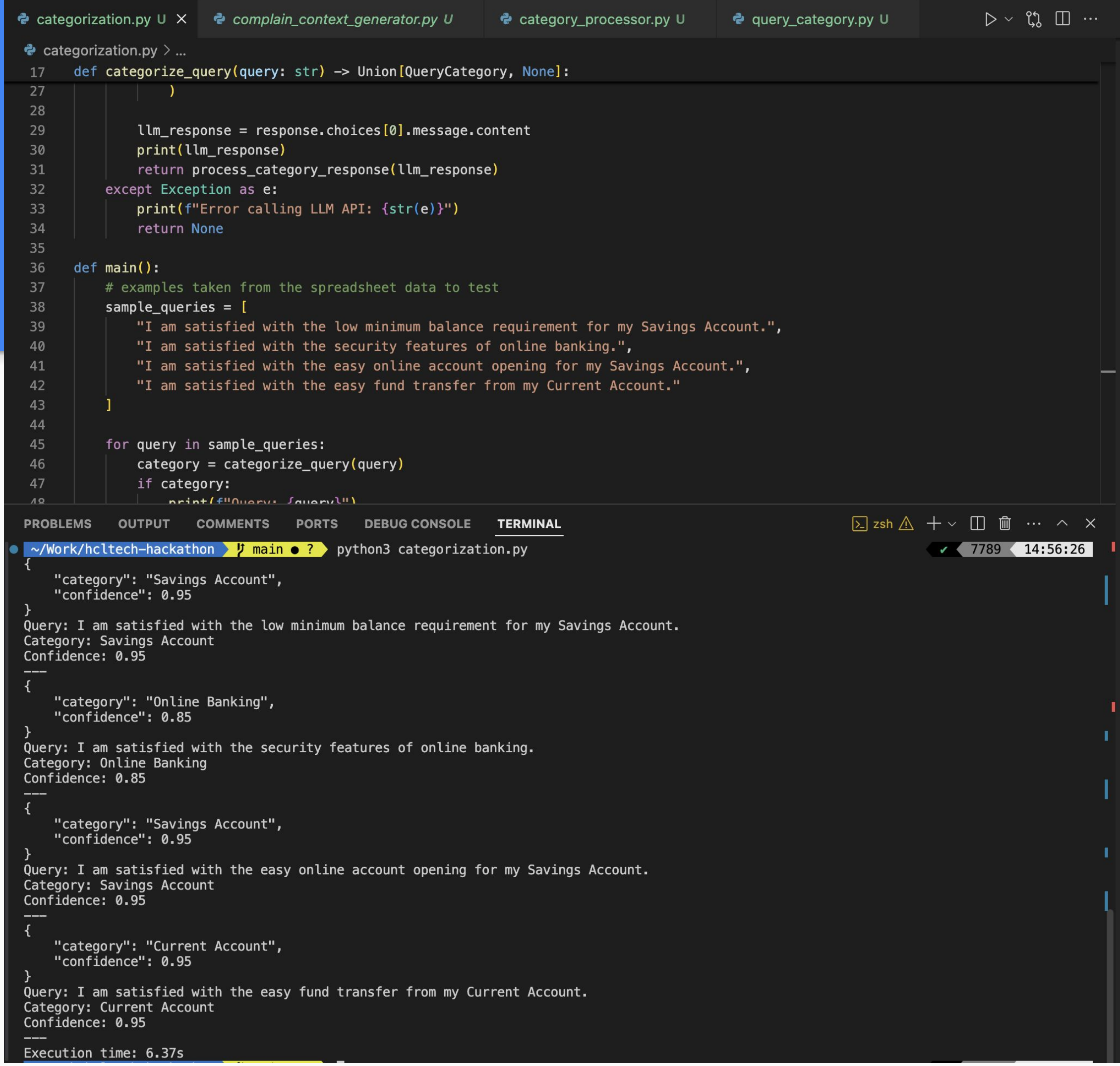
Assistant chatbot

category

I first ensure that the model returns correct category given the complaint feedback from the user.

I use pydantic model to ensure return type of a json as shown.

I use a smaller gpt4o-mini model for it

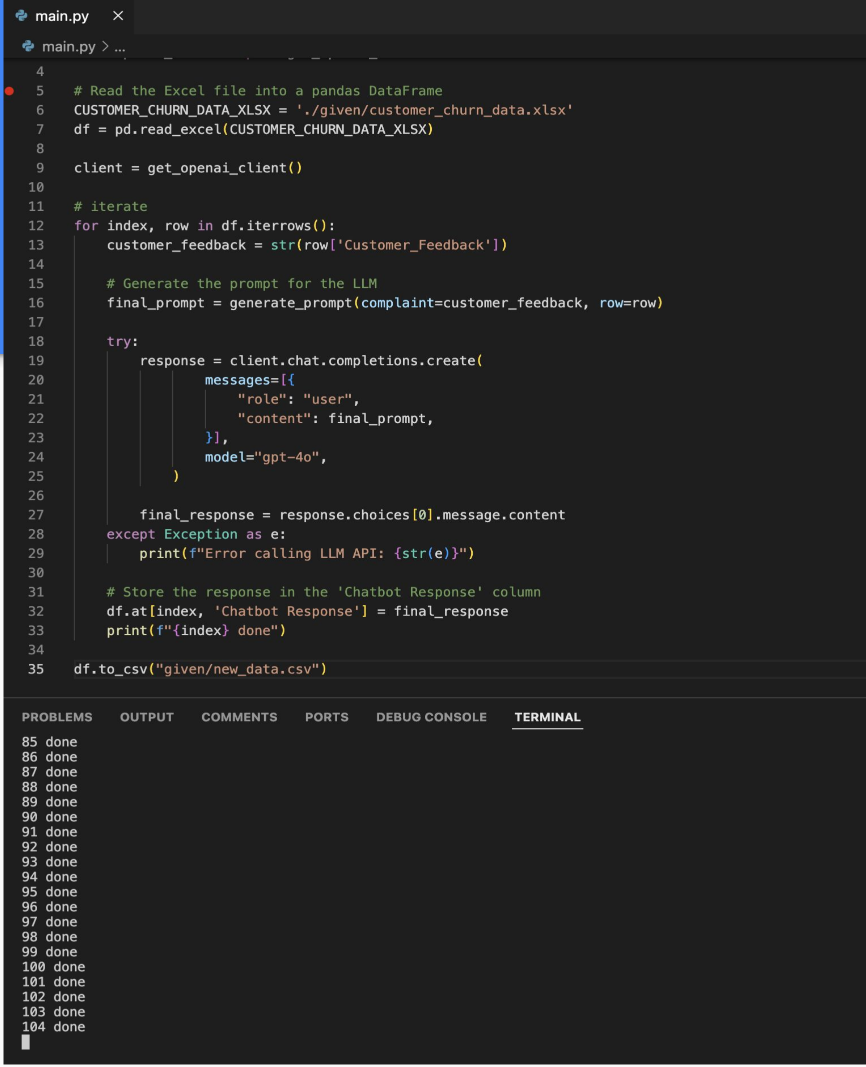


```python
def categorize_query(query: str) -> Union[QueryCategory, None]:
            )

        llm_response = response.choices[0].message.content
        print(llm_response)
        return process_category_response(llm_response)
    except Exception as e:
        print(f"Error calling LLM API: {str(e)}")
        return None

def main():
    # examples taken from the spreadsheet data to test
    sample_queries = [
        "I am satisfied with the low minimum balance requirement for my Savings Account.",
        "I am satisfied with the security features of online banking.",
        "I am satisfied with the easy online account opening for my Savings Account.",
        "I am satisfied with the easy fund transfer from my Current Account."
    ]

    for query in sample_queries:
        category = categorize_query(query)
        if category:
            print(f"Query: {query}")
```

```
~/Work/hcltech-hackathon  main ● ?  python3 categorization.py          7789  14:56:26
{
    "category": "Savings Account",
    "confidence": 0.95
}
Query: I am satisfied with the low minimum balance requirement for my Savings Account.
Category: Savings Account
Confidence: 0.95
---
{
    "category": "Online Banking",
    "confidence": 0.85
}
Query: I am satisfied with the security features of online banking.
Category: Online Banking
Confidence: 0.85
---
{
    "category": "Savings Account",
    "confidence": 0.95
}
Query: I am satisfied with the easy online account opening for my Savings Account.
Category: Savings Account
Confidence: 0.95
---
{
    "category": "Current Account",
    "confidence": 0.95
}
Query: I am satisfied with the easy fund transfer from my Current Account.
Category: Current Account
Confidence: 0.95
---
Execution time: 6.37s
```

Chatbot responses

I use DYNAMIC PROMPT to select data points from the spreadsheet based on the user's query and then use the OpenAI gpt-4o for generating responses.


main.py ×

main.py > ...

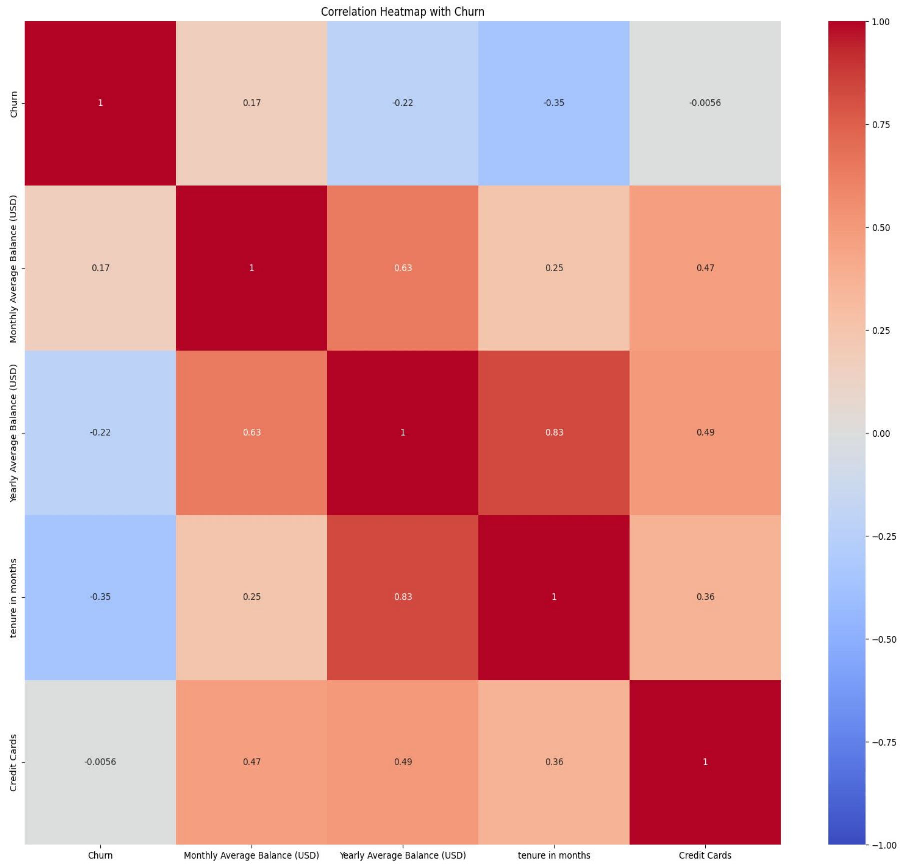
```python

# Read the Excel file into a pandas DataFrame
CUSTOMER_CHURN_DATA_XLSX = './given/customer_churn_data.xlsx'
df = pd.read_excel(CUSTOMER_CHURN_DATA_XLSX)

client = get_openai_client()

# iterate
for index, row in df.iterrows():
    customer_feedback = str(row['Customer_Feedback'])

    # Generate the prompt for the LLM
    final_prompt = generate_prompt(complaint=customer_feedback, row=row)

    try:
        response = client.chat.completions.create(
                messages=[{
                    "role": "user",
                    "content": final_prompt,
                }],
                model="gpt-4o",
            )

        final_response = response.choices[0].message.content
    except Exception as e:
        print(f"Error calling LLM API: {str(e)}")

    # Store the response in the 'Chatbot Response' column
    df.at[index, 'Chatbot Response'] = final_response
    print(f"{index} done")

df.to_csv("given/new_data.csv")
```

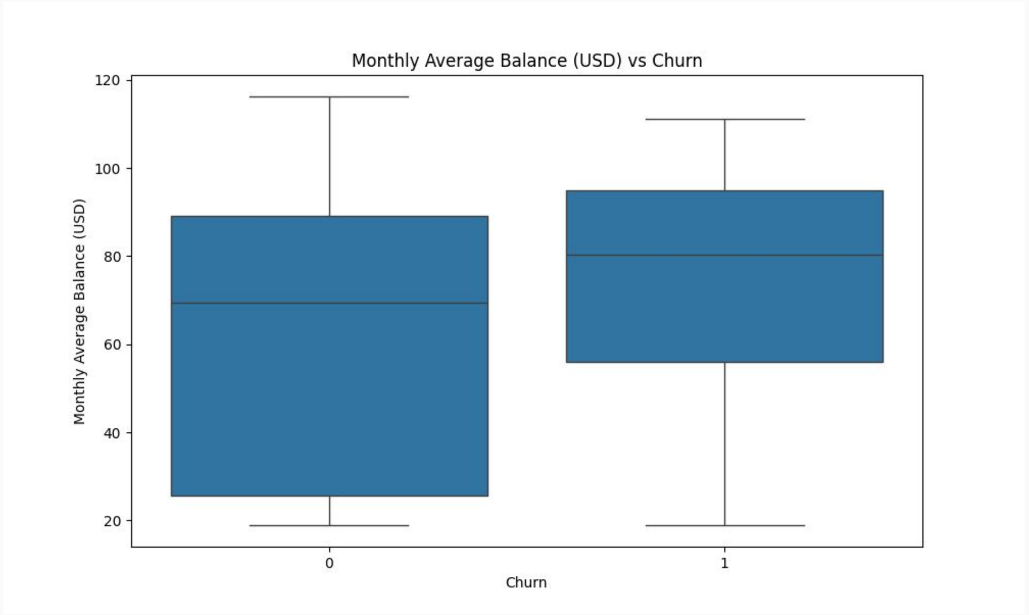
PROBLEMS OUTPUT COMMENTS PORTS DEBUG CONSOLE TERMINAL

```
85 done
86 done
87 done
88 done
89 done
90 done
91 done
92 done
93 done
94 done
95 done
96 done
97 done
98 done
99 done
100 done
101 done
102 done
103 done
104 done
```

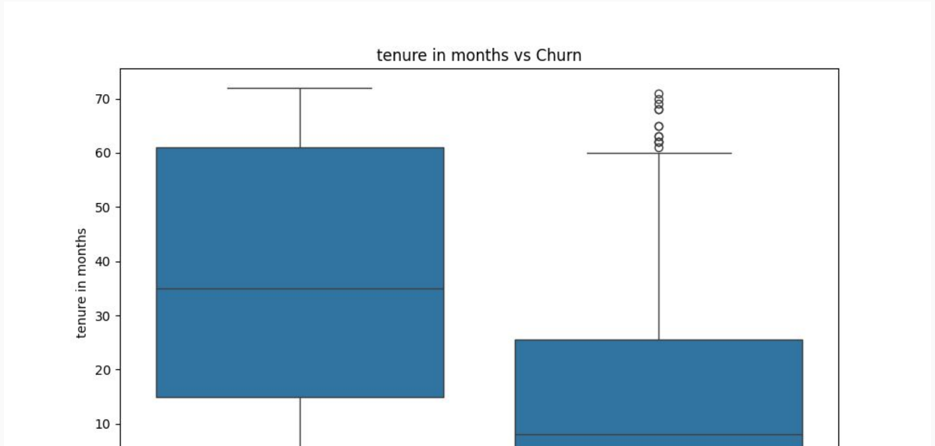
churn

Since churn is already given as a yes/no in the dataset.. So I decided to correlate it with the most relevant factors eg. credit cards, monthly & yearly average balances.




Monthly Average Balance (USD) vs Churn
120
100
80
60
40
20
Monthly Average Balance (USD)
0
1
Churn

more plots


tenure in months vs Churn
70
60
50
40
30
20
10
tenure in months

Thanks!