Midterm - Initial report - Team FinSavvys

Project Goals

Our project's goal to develop an **AI driven Financial chatbot** improves people's understanding and money management. It shows users where their money is going, reads transaction records and classifies spending into basic categories. By doing this, the chatbot helps users make better spending decisions and lessens the burden of tracking and budgeting.

Most people know they should keep track of their spending, but they rarely find the time to do it. They are long, hard to read and cover a lot of different accounts. It's easy to lose sight of where money is going because of this. That's what our system tries to fix by turning raw transaction data into insights that are clear and simple to read. It saves time and helps people notice habits they might not have noticed before. This makes budgeting easier and less stressful.

The chatbot is meant for anyone who wants a clearer picture of their daily spending - students, young professionals or anyone learning to budget. It helps by automatically sorting expenses, answering basic money questions and pointing out spending patterns. The main benefit is convenience: users don't need to go through statements manually and they get quick feedback on how to manage their money better.

Basic Approach

We are building the chatbot in small steps so each part can be tested before adding new features. Firstly, we will prepare clean sample data and build a simple model that can identify and simply label different types of transactions. After that, we will connect it to the chatbot interface. This will help users to ask questions and get answers. Once the basic version works well, we will add more functions like monthly summaries and spending advice. Working in stages helps us spot problems early and improve as we go.

We will use a few different AI methods together:

Natural Language Processing: so the chatbot can understand what the user is asking

Machine Learning: to group expenses automatically and adjust from feedback

Knowledge Representation: to store and organize user data for quick look-ups

Pattern Recognition: to find trends in spending and generate small suggestions

Everyone on the team is involved in both the coding and research parts, but each of us also takes charge of specific areas. **Sudeepta and Rucha** focus more on research and documentation to keep our design clear and consistent. **Mitali** works on connecting research ideas to the program itself. **Sasi** handles much of the main development and integration work. **Sahit** takes care of testing, demos and checking that each build runs smoothly. We meet every week to review what's done, discuss issues and plan the next small goal. This way, progress stays steady and the workload is balanced for everyone.

Anticipated results

The Financial Chatbot is on track to help people change their way of tracking and understanding about their daily finances by giving them a more intelligent, engaging and easily reachable financial bot. This section about anticipated results talks mainly around 2 focused points, the first one being the **functional results and expectations** that talks about what the system will perform and the second one being the **performance measures** that talks about how reliably and accurately the system will operate error-free.

From a **functional point-of-view**, the system will be developed to autonomize the way of monitoring the finances/expenses and its capability to make effective judgements. It will actually allow all the users to upload their financial bank statements or transaction histories, in formats that are highly accessible and acceptable like in the form of

CSV, Excel or pdf. Once the documents get uploaded in the system, it will be able to automatically pull and classify the desired transactions into some predefined categories such as the food, housing, transportation, health, etc. This will actually help the users to avoid doing the long-hours manual and boring classification or doing some end-month analysis on transactions with the help of spreadsheets.

Apart from the transaction classification, the chatbot will generate analysis based on patterns related to the users' spending history. This will be implemented by detecting some trends and repeatable expenditures, emphasizing the areas where the user is spending more and offering some useful insights about the user's transaction patterns. Another of the functional results is the highly engaging process of answering to user's questions, where a user may ask some usual questions like "How much did I spend on food last month?", "In which category did I spend the most, last month?" and so on. The user will be receiving correct and generic responses . The chatbot will also have the ability to generate personalized budgets month-wise on the basis of previous transaction records. Over time, it will efficiently use the feedback provided by the user to continue refining its level of understanding and replies, signifying a clear and focused element of knowledge acquiring and developing skills.

In terms of **performance measures**, the chatbot is expected to provide a high amount of real-time responses, correctness and its simplicity to use the system. The system should be able to work on huge volumes of transactional data with ease, without any potential lags. Its correctness in categorizing transactions and providing useful information will continue to enhance progressively with the help of the combination of supervised and reinforcement learning strategies. Efficiency will be considered by the ability of the system to inspect and reply in almost real time. Usability is also considered to be a major aspect, the chatbot's interface (if time permits) will be a simple, user-friendly and accessible one, making sure of complete accessibility for the intended users irrespective of their technical experience. In the coming times, we have also detected a potential use case to include the feature of dialogue text and interactivity based on voices that will in turn make the system more interactive and responsive.

The success of the system will be rated based on several factors:
-> The measure of the transactions that are accurately categorized in comparison against a dataset that has been validated to make sure of the system's correctness rate.
-> The time that will be taken by the system to understand the uploaded statements and replies generated to the queries.
-> The satisfaction to the user that will be measured through some potential feedback process, focusing on the system's readability, user assistance and the ease of engaging.
-> The reliability of the system will be measured through the accuracy of generated transaction patterns and the budget overview.
-> Whether the knowledge acquired is improving or not will be measured through the system's improved correctness and its quality of replies as more users' financial data and feedback are considered in a combined form.

When completely implemented, these factors will actually confirm that the chatbot system successfully removes the burden of manual management of personal finances, giving more meaningful information based on data that will help the users take better decisions on their finances.

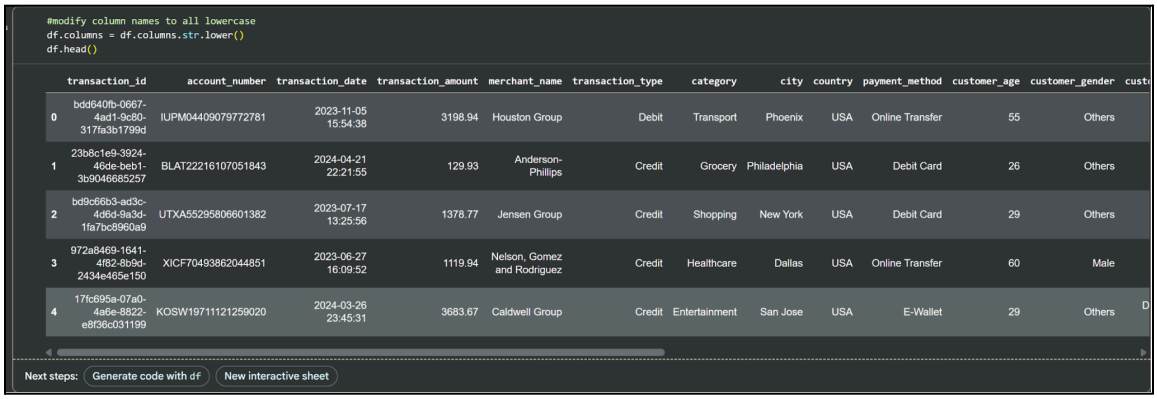
Current Implementation Status

Our team has made noticeable progress in the stages concerning preparation of data and preprocessing of data, which actually is treated as the foundation of the system's capabilities concerning analysis. Necessary scripts are developed to read financial transaction based datasets, cleaning the dataset and then changing the raw records into a more structured format. This pretty much ensures that consistency of the data remains, and no missing values and irregularities are faced in the later stage of the code when the analysis actually begins. These preliminary steps are required to validate for the correctness and reliability of any financial analysis driven by AI.

A portion of our code snippet related to **Data EDA**:

```
import pandas as pd
import numpy as np
```

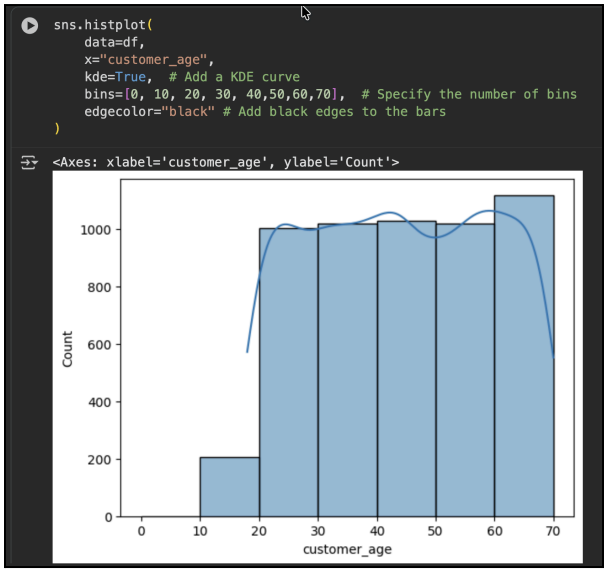
```
import matplotlib.pyplot as plt
import seaborn as sns

#modify column names to all lowercase
df.columns = df.columns.str.lower()
df.head()
```



	transaction_id	account_number	transaction_date	transaction_amount	merchant_name	transaction_type	category	city	country	payment_method	customer_age	customer_gender	cust
0	bdd640fb-0667-4ad1-9c80-317fa3b1799d	IUPM04409079772781	2023-11-05 15:54:38	3198.94	Houston Group	Debit	Transport	Phoenix	USA	Online Transfer	55	Others	
1	23b8c1e9-3924-46de-beb1-3b9046685257	BLAT22216107051843	2024-04-21 22:21:55	129.93	Anderson-Phillips	Credit	Grocery	Philadelphia	USA	Debit Card	26	Others	
2	bd9c66b3-ad3c-4d6d-9a3d-1fa7bc8960a9	UTXA55295806601382	2023-07-17 13:25:56	1378.77	Jensen Group	Credit	Shopping	New York	USA	Debit Card	29	Others	
3	972a8469-1641-4f82-8b9d-2434e465e150	XICF70493862044851	2023-06-27 16:09:52	1119.94	Nelson, Gomez and Rodriguez	Credit	Healthcare	Dallas	USA	Online Transfer	60	Male	
4	17fc695a-07a0-4a6e-8822-e8f36c031199	KOSW19711121259020	2024-03-26 23:45:31	3683.67	Caldwell Group	Credit	Entertainment	San Jose	USA	E-Wallet	29	Others	

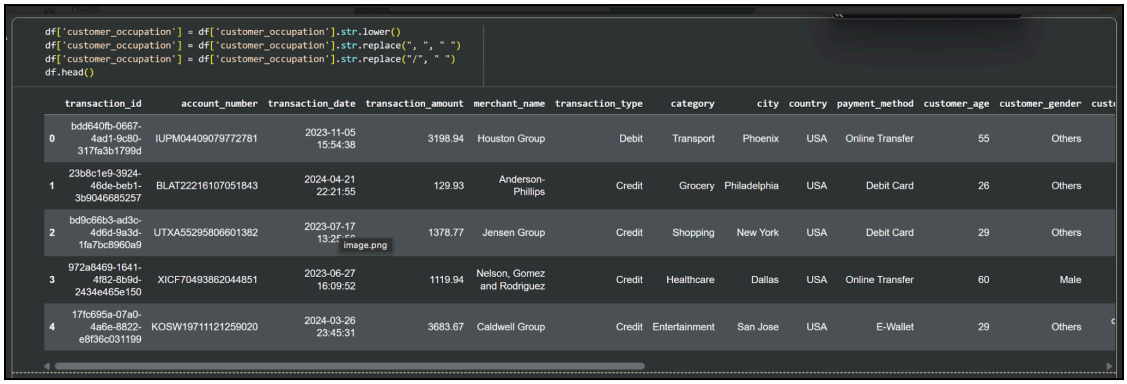
Next steps: Generate code with df | New interactive sheet

```
cust_df = df.groupby("account_number")[['customer_age', 'customer_gender',
'customer_occupation', 'customer_income']].apply(lambda x: x).reset_index()
cust_df.head()

sns.histplot(
data=df,
x="customer_age",
kde=True, # Add a KDE curve
bins=[0, 10, 20, 30, 40,50,60,70], # Specify the number of bins
edgecolor="black" # Add black edges to the bars
```




```
sns.histplot(
data=df,
x="customer_income",
kde=True, # Add a KDE curve
bins=5, # Specify the number of bins
edgecolor="black" # Add black edges to the bars

df['customer_occupation'] = df['customer_occupation'].str.lower()
```

```
df['customer_occupation'] = df['customer_occupation'].str.replace(", ", " ")
df['customer_occupation'] = df['customer_occupation'].str.replace("/", " ")
df.head()
```



	transaction_id	account_number	transaction_date	transaction_amount	merchant_name	transaction_type	category	city	country	payment_method	customer_age	customer_gender
0	bdd640fb-0667-4ad1-9c80-317fa3b1799d	IUPM04409079772781	2023-11-05 15:54:38	3198.94	Houston Group	Debit	Transport	Phoenix	USA	Online Transfer	55	Others
1	23b8c1e9-3924-46de-beb1-3b9046685257	BLAT22216107051843	2024-04-21 22:21:55	129.93	Anderson-Phillips	Credit	Grocery	Philadelphia	USA	Debit Card	26	Others
2	bd9c66b3-ad3c-4d6d-9a3d-1fa7bc8960a9	UTXA55295806601382	2023-07-17 13:2[illegible]	1378.77	Jensen Group	Credit	Shopping	New York	USA	Debit Card	29	Others
3	972a8469-1641-4f82-8b9d-2434e465e150	XICF70493862044851	2023-06-27 16:09:52	1119.94	Nelson, Gomez and Rodriguez	Credit	Healthcare	Dallas	USA	Online Transfer	60	Male
4	17fc695a-07a0-4a6e-8822-e8f36c031199	KOSW19711121259020	2024-03-26 23:45:31	3683.67	Caldwell Group	Credit	Entertainment	San Jose	USA	E-Wallet	29	Others

image.png

```
from sentence_transformers import SentenceTransformer
st_model = SentenceTransformer("all-MiniLM-L6-v2")
X_emb = st_model.encode(
texts,
normalize_embeddings=True,
show_progress_bar=True
)
import numpy as np
print("Embedding matrix shape:", np.array(X_emb).shape)
```

Also, our team has already developed an initial machine learning framework for the categorization of the transaction. So, the code written validates if any column is present in the dataset that has labels, it's because if labels are not there in the dataset, a weak labeling process is applied leveraging a group of rules on the basis of predefined-keywords to allocate approximate classification to the transactions. This actually makes sure that the model starts acquiring knowledge even with the help of partially labelled or completely unlabelled data. The text data is then changed into embeddings and divided into training and testing sets. A Logic Regression model is then trained and tested, with the outcomes shown using classification_report, giving an early measure of performance on the basis of classification.

A portion of our code snippet related to **ML Model for Transaction Classification**:

```
LABEL_COL = "category"
if LABEL_COL in df.columns:
y = df[LABEL_COL].astype(str).values
else:
RULES = {
"Food": ["cafe","coffee","restaurant","mcdonald","ubereats","doordash"],
```

```
"Housing": ["rent","landlord","apartment","zillow","airbnb"],
"Transport": ["uber","lyft","fuel","gas","metro","bus","shell","exxon"],
"Bills": ["utility","electric","water","internet","comcast","verizon"],
"Shopping": ["amazon","walmart","target","etsy","ebay"],
"Health": ["pharmacy","walgreens","cvs","clinic","dental"],
"Subscriptions": ["spotify","netflix","apple","google storage","prime"]
}
def weak_label(s: str) -> str:
s = s.lower()
for lab, keys in RULES.items():
if any(k in s for k in keys): return lab
return "Other"
df["weak_category"] = df[TEXT_COL].apply(weak_label)
LABEL_COL = "weak_category"
y = df[LABEL_COL].values
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

X_train, X_test, y_train, y_test = train_test_split(
    X_emb, y, test_size=0.2, random_state=42, stratify=y
)

clf = LogisticRegression(max_iter=1000, n_jobs=None)
clf.fit(X_train, y_train)
print(classification_report(y_test, clf.predict(X_test)))
```

```
                precision    recall  f1-score   support

      Clothing       0.11      0.11      0.11        79
     Education       0.03      0.04      0.04        78
   Electronics       0.01      0.01      0.01        76
 Entertainment       0.08      0.10      0.09        82
       Fitness       0.08      0.09      0.09        75
          Food       0.07      0.06      0.06        71
       Grocery       0.06      0.06      0.06        78
    Healthcare       0.06      0.04      0.05        74
       Housing       0.11      0.12      0.11        69
       Savings       0.10      0.09      0.09        78
      Shopping       0.05      0.04      0.05        78
     Transport       0.06      0.05      0.06        73
        Travel       0.09      0.12      0.10        83
     Utilities       0.05      0.06      0.06        84

      accuracy                           0.07      1078
     macro avg       0.07      0.07      0.07      1078
  weighted avg       0.07      0.07      0.07      1078
```

By incorporating both structured data management and foundational predictive abilities, the system's functionality to classify the transactions correctly is well showcased. We have also started our work for the next phase of the project.

To be Implemented

In the pipeline, we still have the implementation of Natural language query processing, autonomous budget generation and continuous feedback learning. These components will actually allow the system to understand the user's wish, generate some personalized insights and continue to enhance based on engagements. Upcoming work will actually focus on combining the AI modules to shift the chatbot system from foundational transaction categorization to intelligent judgement taking system.

Bibliography

Kobets, V. M., & Kozlovskyi, K. H. (2022). **Application of chat bots for personalized financial advice**. *Herald of Advanced Information Technology, 5*(3), 229–242. https://doi.org/10.15276/hait.05.2022.18

Lu, Q., Luo, Y., Zhu, L., Tang, M., Xu, X., & Whittle, J. (2023). **Developing responsible chatbots for financial services: A pattern-oriented responsible artificial intelligence engineering approach.** *IEEE Intelligent Systems, 38*(6), 42–51. https://doi.org/10.1109/MIS.2023.3320437

Hwang, S., & Kim, J. (2021). **Toward a chatbot for financial sustainability**. *Sustainability, 13*(6), 3173. https://doi.org/10.3390/su13063173