Anti-Money Laundering Analysis using Large Language Models

Kanya Krishi
Kishorekumar Suresh

11/28/2023

Agenda

1. Introduction
2. What are LLMs
3. Types of LLMs used
4. LLM results
5. ML models used
6. ML results
7. Analysis
8. Retrieval Augmented Generation
9. Conclusion
10. Task Allocation

Introduction

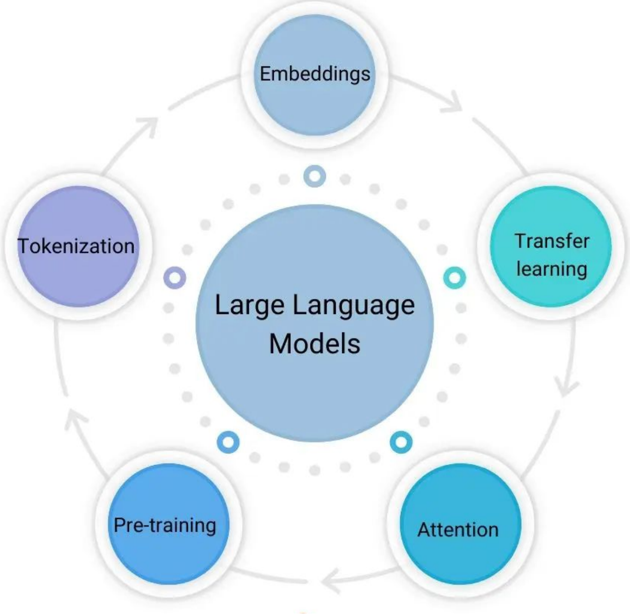
- Dataset used - IBMs Anti-Money Laundering
- Testing data - 100 (A mix of both labels)
- Data Preprocessing is required
  - To ensure compatibility with model
  - Handle missing or incomplete data
  - Better understanding and interpretation of data

Transaction occurred on [Timestamp] from bank [From Bank] with originating account [Originating Account]. The transaction was directed to bank [To Bank] with receiving account [Receiving Account]. The amount received was [Amount Received] [Receiving Currency], and the amount paid was [Amount Paid] [Payment Currency]. The payment method used was [Payment Format]. This transaction is flagged as [Is Laundering: 'Laundering'/'Not Laundering']

Large Language Models

- They are used for Natural Language processing and built using deep learning techniques especially neural networks.
- Used for:
  - Natural language understanding and generation.
  - They provide more relevant and coherent responses or outputs.
  - They can be fine-tuned for a specific tasks.

How does **LLM** work? Key building blocks



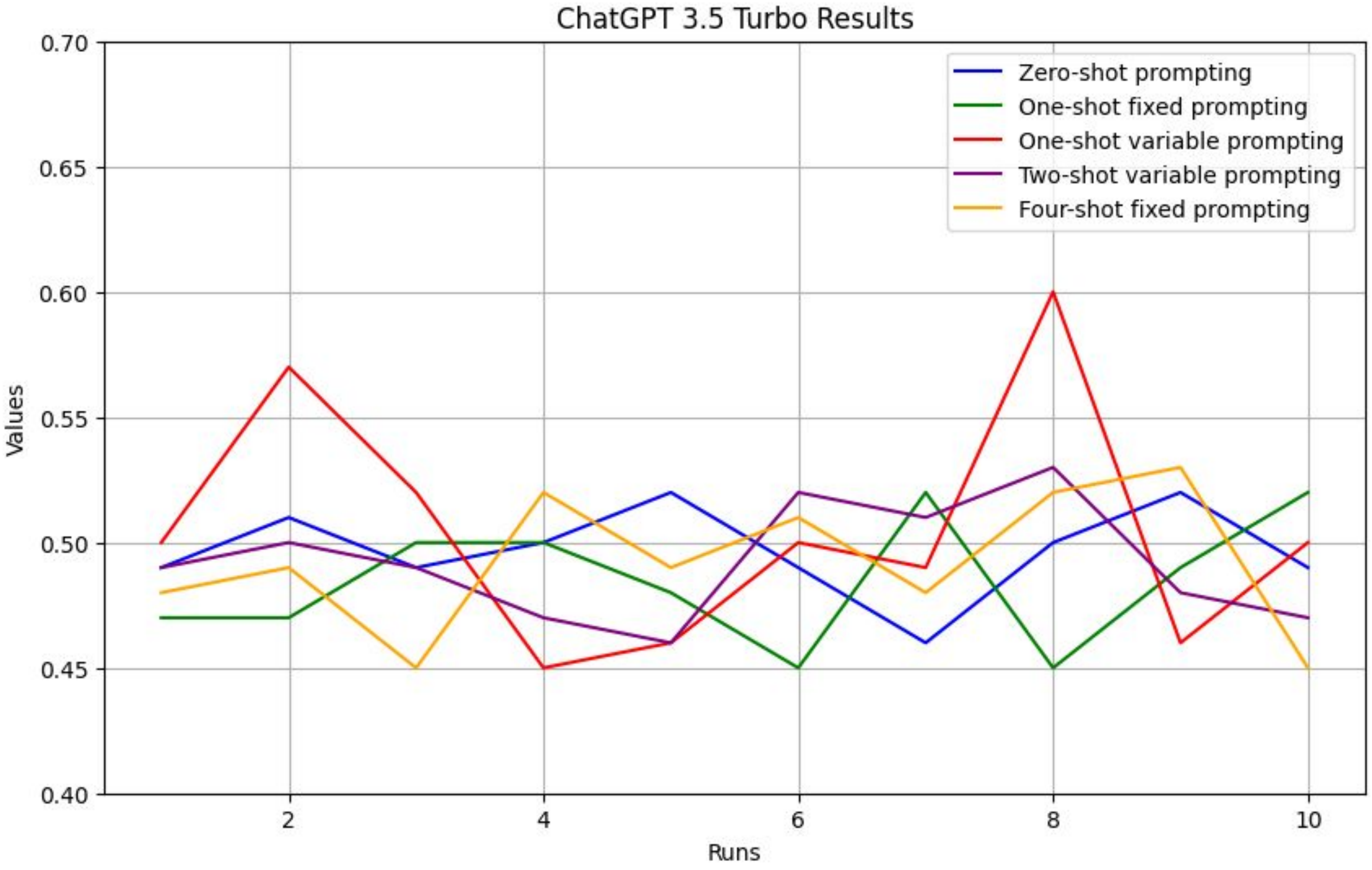
Types of LLMs

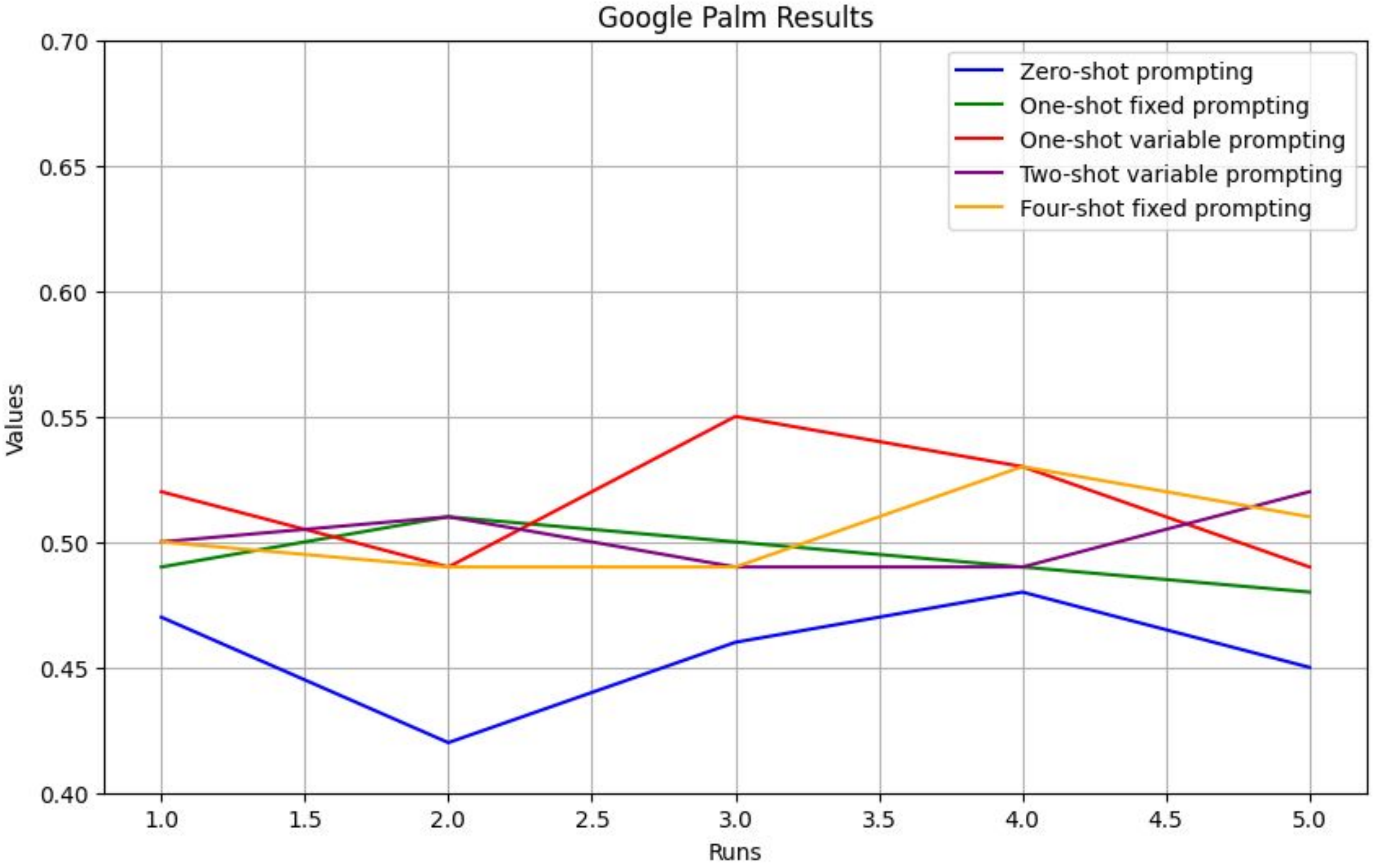
1. GPT-3.5 (OpenAI)
   a. Scalable
   b. Advance language understanding and generation.
2. PaLM (Google)

LLM Analysis

We ran the test data on both LLMs for the following cases:

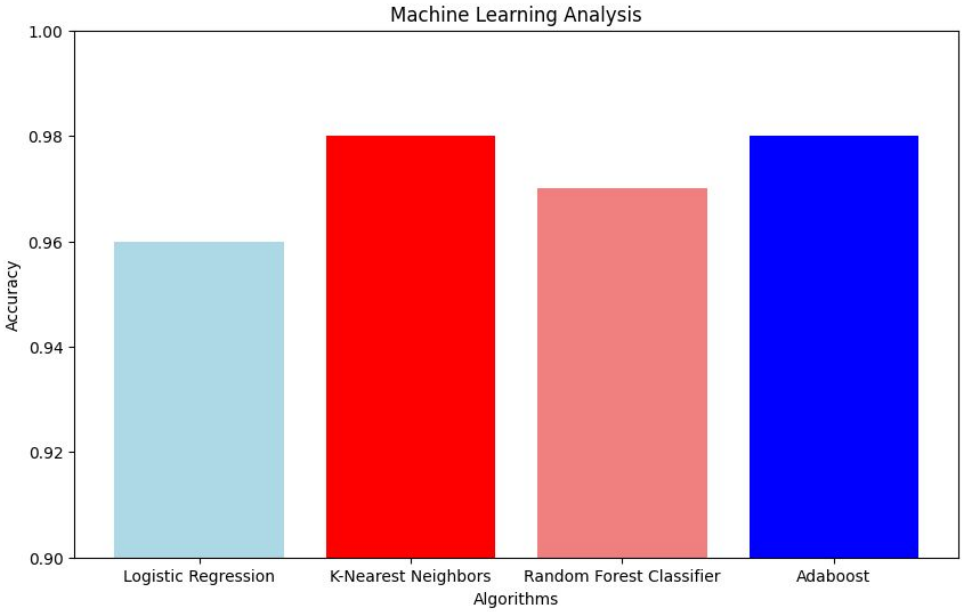
1. Zero-shot prompting
2. One-shot prompting
   a. Fixed
   b. Variable
3. Two-shot variable prompting
4. Four-shot fixed prompting


ChatGPT 3.5 Turbo Results
Zero-shot prompting
One-shot fixed prompting
One-shot variable prompting
Two-shot variable prompting
Four-shot fixed prompting
Values
0.70
0.65
0.60
0.55
0.50
0.45
0.40
Runs
2
4
6
8
10


Google Palm Results
Zero-shot prompting
One-shot fixed prompting
One-shot variable prompting
Two-shot variable prompting
Four-shot fixed prompting
Values
0.70
0.65
0.60
0.55
0.50
0.45
0.40
Runs
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

Machine Learning Models

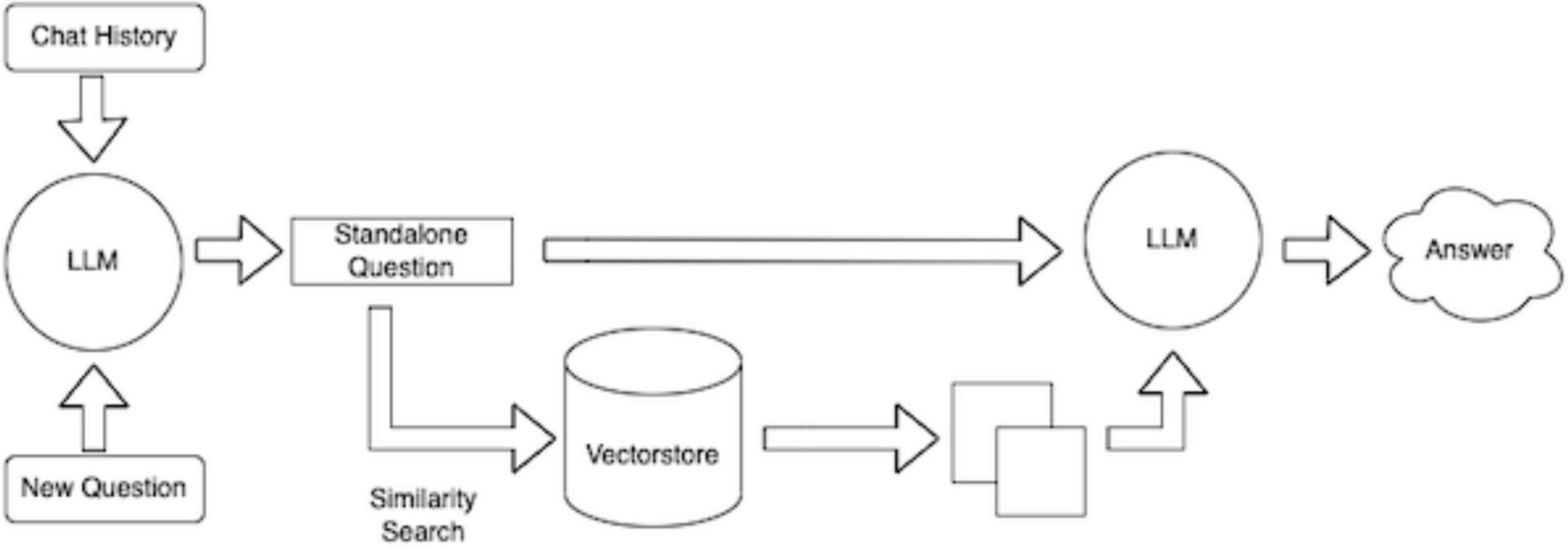
- They are better suited for classification tasks - learn complex patterns.
- Models used - 'Logistic Regression', 'K-Nearest Neighbors', 'Random Forest Classifier', 'Adaboost'.

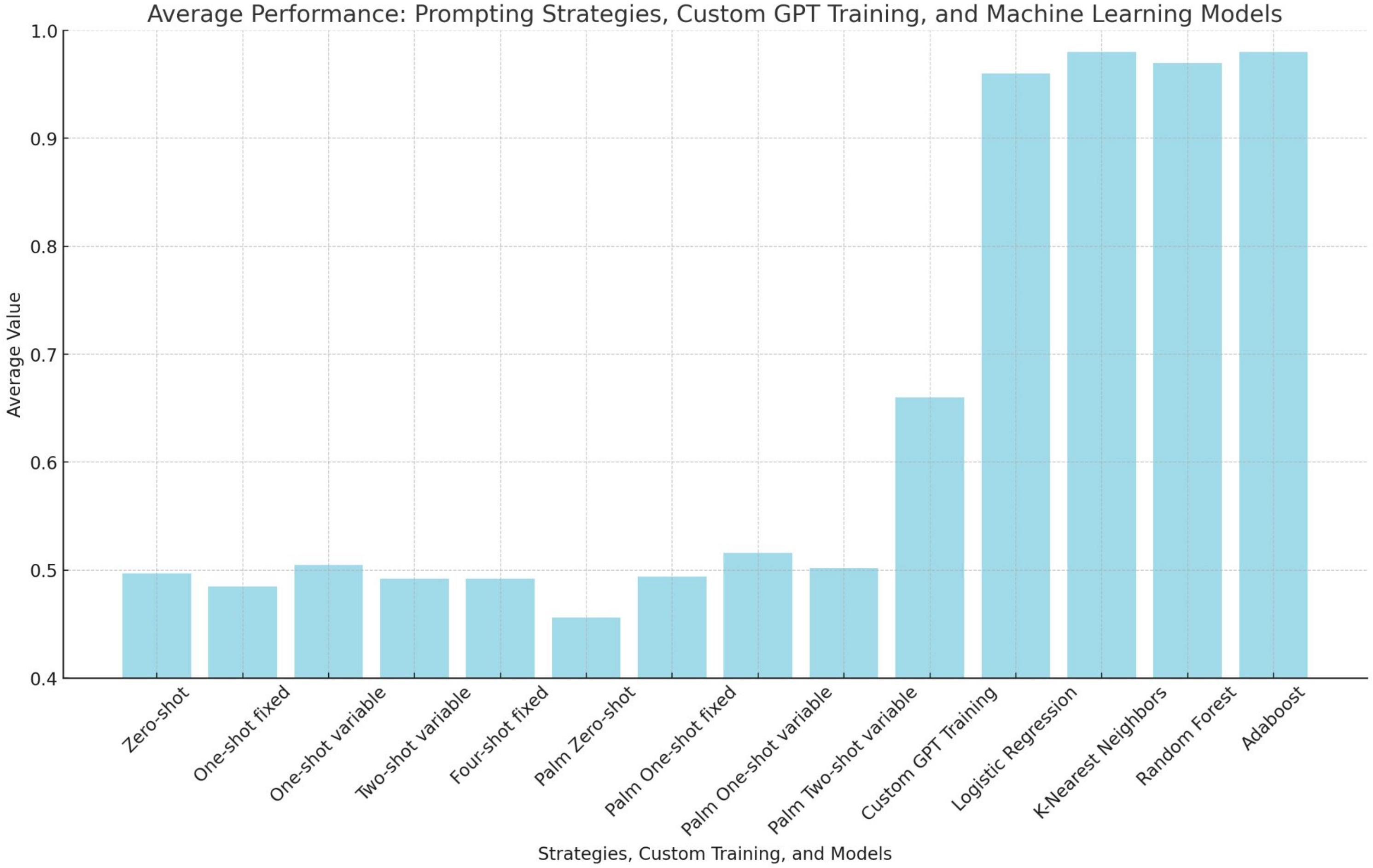


Retrieval Augmented Generation

- Langchain package is used for Retrieval Augmented Generation
- Retrieval methods, such as semantic search, involve calculating numerical vectors for documents and storing them in a vector database.
- Queries are also vectorized, and documents closest to the query in embedding space are retrieved.
- A diagram is provided below to illustrate this retrieval process.

Retrieval Augmented Generation




Average Performance: Prompting Strategies, Custom GPT Training, and Machine Learning Models
Average Value
1.0
0.9
0.8
0.7
0.6
0.5
0.4
Zero-shot
One-shot fixed
One-shot variable
Two-shot variable
Four-shot fixed
Palm Zero-shot
Palm One-shot fixed
Palm One-shot variable
Palm Two-shot variable
Custom GPT Training
Logistic Regression
K-Nearest Neighbors
Random Forest
Adaboost
Strategies, Custom Training, and Models

Conclusion

- In conclusion, LLM models are better suited for natural language tasks such as Sentiment Analysis, Email Spam classification to name a few
- Machine Learning models perform better for Money Laundering Classification because they can identify patterns in the dataset with limited features too
- LLM can perform better if we can provide more features and information like transaction history, account information and other fields
- RAGs performed slightly better as similar examples based on the inputs are retrieved and then passed to the prompt

Task Allocation

- Kanya Krishi’s Tasks
  - GPT 3.5 Turbo API
  - Machine Learning Analysis
  - Presentation Slides
- Kishorekumar Suresh’s Tasks
  - Google PaLM API
  - Retrieval Augmented Generation - RAG
  - Presentation Slides

Thank You