### **FinAdvisor - Financial Advisory Tool for contextually-aware personal finance assistance DS5500: Data Science Capstone**

Prof. Kylie A. Bemis

P2 Project Report[[Github](https://github.com/sriksven/FinAdvisor_LLM)] **Team 2:** Laasya Anantha Prasad, Krishna Venkatesh, Rakshak Kunchum

### **1. Abstract:**

The FinAdvisor project introduces an AI-driven financial advisory tool designed to provide real-time, contextually-aware personal finance assistance. Traditional financial advisory services and financial human experts often lack personalization, timeliness, and accessibility—limitations FinAdvisor addresses by leveraging Large Language Models (LLMs) fine-tuned with financial datasets like the Financial Alpaca Dataset and FinTalk-19k Dataset sourced for financial question-answer pairs which are used for LLM model fine-tuning.

Using methods like Low-Rank Adaptation (LoRA) and Quantized LoRA (QLoRA), the fine-tuned LLaMA 7B model and the Mistral 7B optimizes performance while maintaining computational efficiency. The system integrates a RESTful API and Gradio-based user interface to ensure seamless user interaction. Evaluation includes metrics such as perplexity and custom heuristic approaches—clarity, specificity, and relevance—to assess response quality. Initial results show promising performance, with the LLaMA model achieving a perplexity score of 7.12, the Mistral model achieving a perplexity of 8.5 and the heuristic metrics effectively measuring response clarity, specificity, and relevance.

FinAdvisor combines AI models with user-friendly technology to democratize financial advisory services, offering real-time, personalized financial advice worldwide.

**2. Introduction:**The financial advisory landscape faces significant limitations with traditional methods, which typically involve human experts. These methods, such as static knowledge bases and in-person consultations with financial human experts, often require clients to visit during limited business hours and rely on outdated, static data. This reliance on human input and physical availability makes traditional advisory services inefficient and inaccessible for timely financial decision-making.

To address these challenges, the FinAdvisor project introduces an AI-powered financial advisory tool leveraging Large Language Models (LLMs) for contextually-aware, real-time financial assistance. LLMs overcome these limitations by providing universal accessibility, real-time insights, and personalized guidance based on the latest market data and individual user needs.

FinAdvisor employs Parameter-Efficient Fine-Tuning (PEFT) techniques such as LoRA (Low-Rank Adaptation) and Quantized LoRA (QLoRA) to adapt pre-trained models like LLaMA 7B and Mistral 7B efficiently, without requiring extensive computational resources. LoRA fine-tunes a subset of model parameters for task adaptation, while QLoRA quantizes weights to 8-bit precision, maintaining model performance while optimizing computational efficiency (Houlsby et al., 2021; Dettmers et al., 2022).

Additionally, FinAdvisor integrates contextual embeddings using Sentence Transformers (Reimers & Gurevych, 2019) to process user queries against financial datasets such as the Financial Alpaca Dataset and FinTalk-19k Dataset. Retrieval is supported by Qdrant, a scalable vector search framework, that enables real-time semantic matching for user inquiries.

This innovative combination of PEFT methodologies, contextual embeddings, and semantic retrieval allows FinAdvisor to provide efficient, scalable, and personalized financial insights. This approach addresses the gaps left by traditional advisory methods, offering dynamic, real-time, and user-tailored financial decision-making support.

**3. Dataset:**The **Financial Alpaca Dataset**[1] combines data from the Stanford Alpaca dataset, the Financial Question and Answer (FIQA) dataset, and additional synthetic data generated by GPT-3.5 and hosted on hugging face[2]. With approximately 68,000 entries, it covers topics such as investment advice, market trends, savings strategies, and personal finance management. The inclusion of synthetic financial scenarios allows the model to learn from diverse and complex financial cases.

The **FinTalk-19k Dataset**, sourced from Reddit discussions, consists of 19,000 entries categorized into Personal Finance, Financial Information, and Public Sentiment. It captures real-world financial concerns, consumer behaviors, and public sentiment, including budgeting, savings strategies, and investment practices. This dataset is particularly valuable for building models that can interpret user concerns and financial intentions effectively.

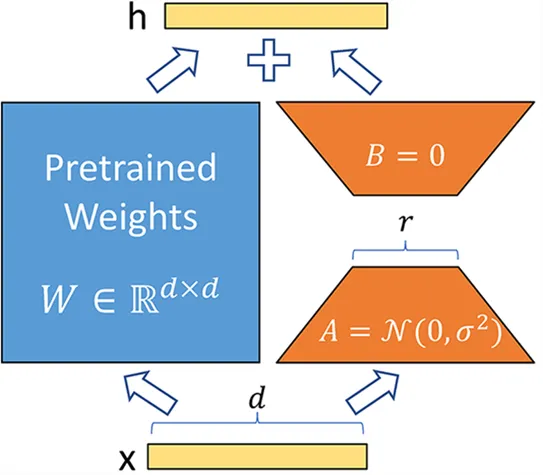
Preprocessing steps included the removal of extraneous white spaces, dashes, and non-ASCII characters to ensure uniformity. This was a crucial step in mitigating noise and optimizing the data for fine-tuning large language models. The cleaned data was further analyzed through embedding analysis using the **all-MiniLM-L6-v2** variant of the Sentence Transformer model. This embedding model was selected for its efficiency in distilling textual data into meaningful representations, facilitating the extraction of financial context essential for downstream tasks.

**Exploratory Data Analysis (EDA)** was performed to understand relationships between key dataset features. The analysis revealed that the lengths of instructions, contexts, and responses exhibited minimal correlation, with a Pearson correlation coefficient of -0.17. This finding demonstrated that the verbosity of user input or context did not constrain the model’s ability to generate concise yet comprehensive responses. Such independence ensures that the model can provide clear and succinct financial advice, irrespective of input complexity, thereby enhancing the user experience.

**4. Methods**

**Fine-Tuning Large Language Models (LLMs)**Fine-tuning LLMs with billions of parameters, such as a 7-billion-parameter model, is computationally intensive and resource-demanding. To address these challenges, **Low-Rank Adaptation (LoRA)**[3]and its quantized extension, **Quantized Low-Rank Adaptation (QLoRA)**[4], offer efficient solutions.

LoRA introduces adapter blocks that reduce the dimensionality of transformer weight matrices. Specifically, it decomposes a weight matrix W of dimensions d×d into two smaller matrices: A with dimensions d×r and B with dimensions r×d, where r is the LoRA rank. This method enables the model to adapt to task-specific data by fine-tuning only the weights of the adapter blocks while freezing the original transformer weights. The use of two hyperparameters—LoRA rank (r) and LoRA alpha (𝛂)—provides control over the size of these reduced matrices, balancing computational efficiency and model adaptability. By reducing the number of trainable parameters, LoRA significantly lowers memory and compute requirements during training.

  
Fig 1: LoRA Weight Matrix decomposition ([QLORA: Efficient Finetuning of Quantized LLMs](https://arxiv.org/pdf/2305.14314))

QLoRA enhances efficiency further by quantizing weight parameters within transformer blocks from 32-bit floating-point precision to 8-bit precision. This quantization drastically reduces the memory footprint and computational load, allowing for faster training without compromising performance. In this project, 8-bit quantization was employed alongside LoRA to fine-tune the LLaMA 7B and Mistral 7B models for a financial advisory context. By combining LoRA’s low-rank parameterization with QLoRA’s precision optimization, the fine-tuning process achieved substantial reductions in training time and resource consumption.

The synergy of LoRA and QLoRA provided a scalable and resource-efficient solution for training LLMs. Freezing the original weights and focusing on lightweight adapter blocks enabled effective specialization without retraining the entire model. Additionally, quantization allowed the training to be performed on standard hardware setups, making the process accessible and practical.

**Models and Inference Pipeline**The fine-tuning of FinAdvisor employed advanced techniques to adapt large language models for specialized financial contexts. The primary approaches used were **LoRA** and **QLoRA**, which allowed efficient adaptation of pre-trained models without the need for full retraining.

Two primary models were fine-tuned during the project: **LLaMA 7B** and **Mistral 7B**. **LLaMA 7B**, known for its computational efficiency, completed fine-tuning in 8 hours and demonstrated adaptability to financial contexts. **Mistral 7B**, which leverages sparse attention mechanisms to achieve high performance with reduced resource demands, completed fine-tuning in 12 hours and produced promising results.

The performance of both models was monitored during training using **Weights & Biases (wandb)**[5], an experiment management tool. Metrics such as training loss, accuracy, and perplexity were tracked in real-time to ensure a systematic and transparent development process.

**Inference Pipeline**The Inference Pipeline integrates a RESTful API and Gradio-based UI to enable real-time interaction with the model. It integrates backend and frontend components to deliver a seamless user experience. A **RESTful API** was developed to serve as the communication bridge between the fine-tuned models and the user interface. It acts as the backbone of the Inference Pipeline, serving as a communication bridge between the fine-tuned language models and the user interface. The API processes user queries, sends them to the models for inference, and returns tailored financial advice. Once a response is generated, the API formats and returns the result to the user in real-time. The **user interface**, developed using **Gradio** [6], allows users to input financial queries and receive personalized advice in real-time. The Gradio interface connects directly with the API. This architecture ensures that users receive quick access to high-quality, personalized financial insights.

### **5. Model Evaluation**

The evaluation of FinAdvisor's models began with traditional metrics such as **perplexity** and **ROUGE scores**. Perplexity measures how well a language model predicts the likelihood of a sequence of words, with lower perplexity indicating better fluency and alignment with natural language. However, while perplexity assesses the naturalness of generated text, it does not evaluate critical nuances such as specificity or the correctness of financial advice.

The ROUGE score, which measures the overlap between generated text and ground-truth responses, was initially considered. However, due to the conversational nature of the dataset—which includes informal advice from financial experts on Reddit—ROUGE proved unsuitable for evaluating the model's performance. Even high-quality responses often scored poorly under this metric because conversational phrasing diverged significantly from the structured ground truth. This limitation became evident upon manual review of the model-generated responses, where many were of high quality but scored low on ROUGE.

Recognizing these shortcomings, the team devised custom heuristic metrics tailored to the evaluation of financial advice:

**Clarity Evaluation**  
Clarity measures whether the financial advice is straightforward to understand. Advice that is convoluted or overly technical can confuse users, defeating the purpose of accessible financial guidance. A simple heuristic evaluates clarity by comparing the sentence length of the model-generated response to that of the ground truth. The assumption is that overly long or excessively short sentences tend to reduce clarity. This metric calculates how closely the length of the generated response matches the ground truth. The closer the lengths, the higher the clarity score, assuming that overly verbose or overly terse responses deviate from the expected format. Although this heuristic provides a simplified evaluation, it offers a practical mechanism for determining whether the structural characteristics of a response contribute to user understanding. The metric's strength lies in its ability to highlight deviations from expected sentence structure norms that compromise clarity, thereby enabling assessment of the response's alignment with accessible financial communication.

  
  
Where, : The length of the model-generated response (in words).  
: The length of the ground truth response (in words).

**Specificity Evaluation**Specificity evaluates how well the generated advice addresses the user’s query with precise details. Financial advice should include relevant terms and details to demonstrate an understanding of the query and provide actionable insights. Specificity directly assesses whether the generated response contains relevant and detailed financial terms. A high degree of specificity ensures that the advice moves beyond generic or overly vague responses, instead offering targeted, context-aware recommendations. For instance, responses that reference concepts such as *asset allocation*, *tax brackets*, or other domain-specific financial terms signal a deeper comprehension of the user’s needs and enhance the practical value of the advice. This ensures the model delivers actionable insights rather than generic or vague advice. For example, advice containing terms like "asset allocation" or "tax bracket" demonstrates a deeper understanding and adds practical value to the user.



Where, : The set of financial terms present in the model-generated response.  
: The set of financial terms present in the ground truth response.

#### **Relevance Evaluation** Relevance determines whether the response aligns with the user’s intent and directly addresses the original financial question. Irrelevant advice, even if clear and specific, diminishes the tool’s utility. Cosine similarity, calculated using TF-IDF vectorization, measures the semantic similarity between the ground truth and the model’s response. This method evaluates whether the response captures the essence of the query. Relevance is measured by analyzing the semantic overlap between the user query and the generated response. By using cosine similarity, this metric ensures that the response is contextually appropriate and aligned with the query's intent.



Where, : The TF-IDF vector representation of the model-generated response.  
: The TF-IDF vector representation of the ground truth response.

The custom metrics—clarity, specificity, and relevance—collectively provide a comprehensive evaluation framework. Unlike standard metrics like ROUGE or perplexity, which are not designed for conversational datasets, these heuristics focus on the unique attributes of high-quality financial advice. By addressing clarity, specificity, and relevance, the team can better assess and improve the model's performance, ensuring it meets or exceeds state-of-the-art benchmarks for contextual accuracy and practical utility. These metrics allow for nuanced comparisons with existing tools, highlighting FinAdvisor’s strengths in delivering user-centric, actionable financial insights.  
  
**6. Results**

**Training Results**The training process for the FinAdvisor project demonstrated consistent and effective learning, as evidenced by a steady reduction in training loss from 2.26 to 1.52 over 250 steps. This significant improvement reflects the model's growing ability to understand and respond to complex financial queries. The fine-tuning methodology, which employed Parameter-Efficient Fine-Tuning (PEFT) with Quantized LoRA (QLoRA), was instrumental in achieving this efficiency, allowing for task-specific learning while maintaining computational feasibility. The fine-tuned **LLaMA model** achieved a perplexity score of 7.12. Although this score is slightly above the state-of-the-art benchmark of 4–6 perplexity, it is commendable given the project's computational constraints and demonstrates that the model delivers fluent, coherent, and contextually appropriate responses.

The training progression for the **Mistral 7B model** is depicted by a consistent decrease in training loss, from an initial value of approximately 2.2 to a final value of 1.78 over a span of 250 steps. This pattern signifies effective learning and the model's successful adaptation to the specialized tasks targeted during its fine-tuning phase. Achieving a final perplexity of 8.5.

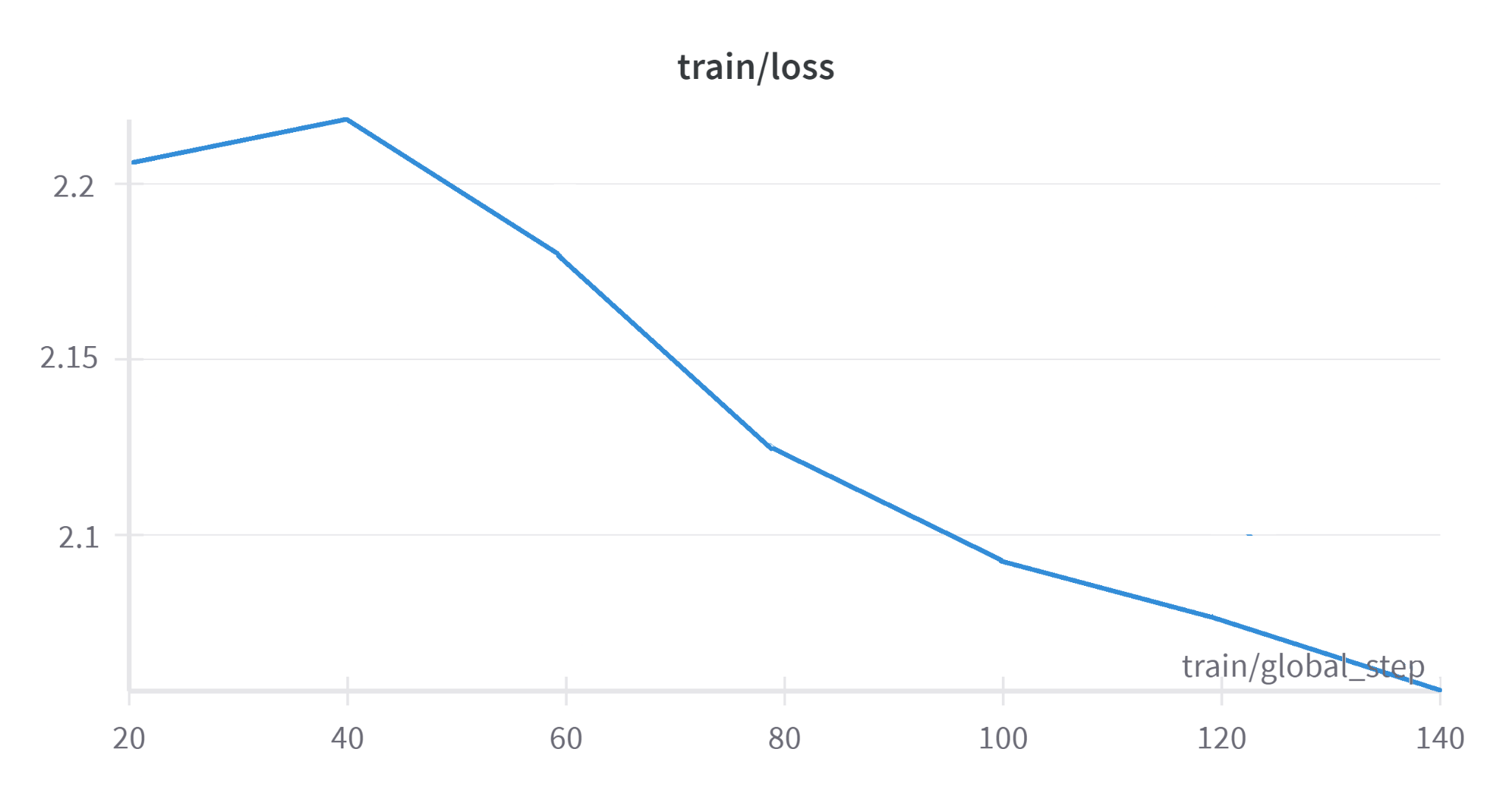
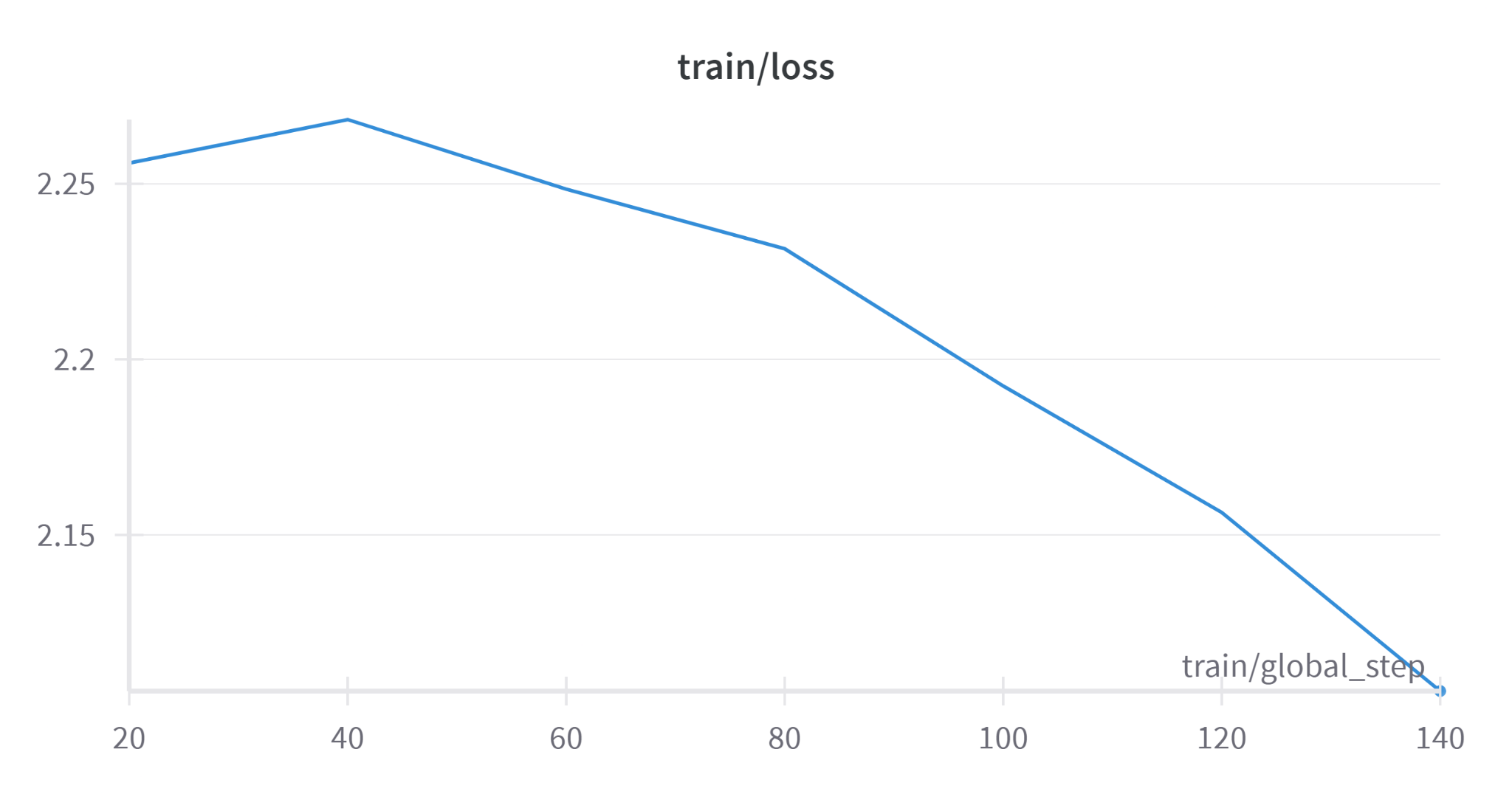


Fig 2: LlaMa 7B fine-tuning training loss Fig 3: Mistral 7B fine-tuning training loss

**Custom Heuristics Evaluation Results**The effectiveness of the model in processing user queries and generating personalized responses was validated through a rigorous evaluation framework. The **fine-tuned LLaMa model** was chosen for **further evaluation** since LLaMa responses were stable and adapted to the context of the instructions very well. While the fine-tuned **Mistral** model provided good context-aware responses but it failed to generate consistent responses for many instructions.

Linguistic metrics such as **ROUGE** are often employed in language modeling projects, they proved less effective for this task due to the conversational and informal nature of the dataset. For example, high-quality financial advice generated by the model often diverged linguistically from the structured ground truth, leading to artificially low ROUGE scores. This limitation was highlighted during manual evaluations of model outputs. To address these shortcomings, custom heuristic metrics were developed and implemented. These metrics focus on three critical aspects of financial advice: **Clarity Evaluation**, **Specificity Evaluation**, and **Relevance Evaluation** offer a more accurate assessment of the conversational accuracy and practical utility of the generated advice. By focusing on these dimensions, the model's performance can be more effectively evaluated and aligned with state-of-the-art systems, ensuring that it meets the high standards required for financial. advisory tools.

The LLaMA fine-tuned model demonstrates an exceptional ability to generate comprehensive and contextually relevant financial responses, further enhancing the evaluation process. By focusing on these dimensions, the model's performance can be more effectively evaluated and aligned with state-of-the-art systems, ensuring that it meets the high standards required for financial advisory tools.

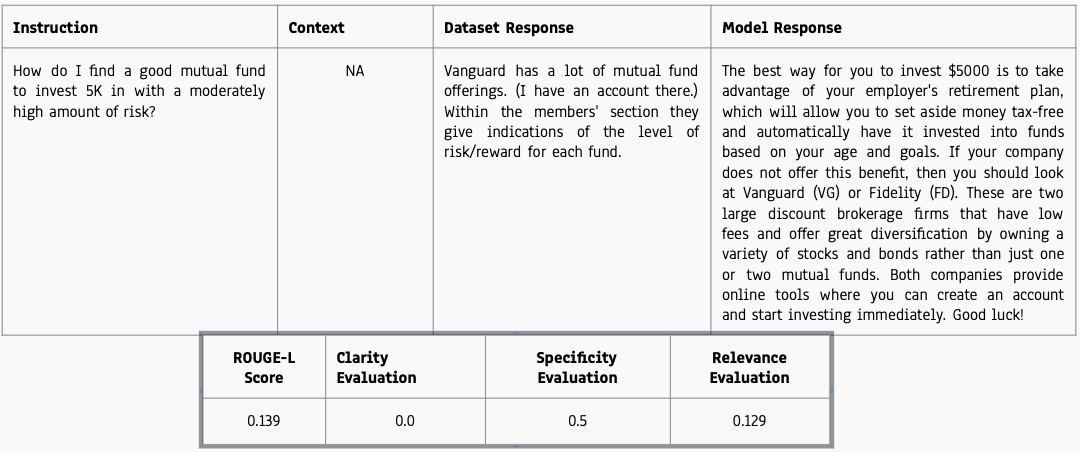


Fig 4: Case-by-case analysis of custom heuristics without context

Clarity is based on a direct comparison of the response length generated by the model and the corresponding ground truth response. When the model's response is significantly longer than the dataset's response, it can lead to reduced clarity, as overly verbose advice risks confusing users. In cases where the model's response is much larger than the ground truth response, the clarity score is calculated as **0**, indicating a lack of conciseness, which is the context.

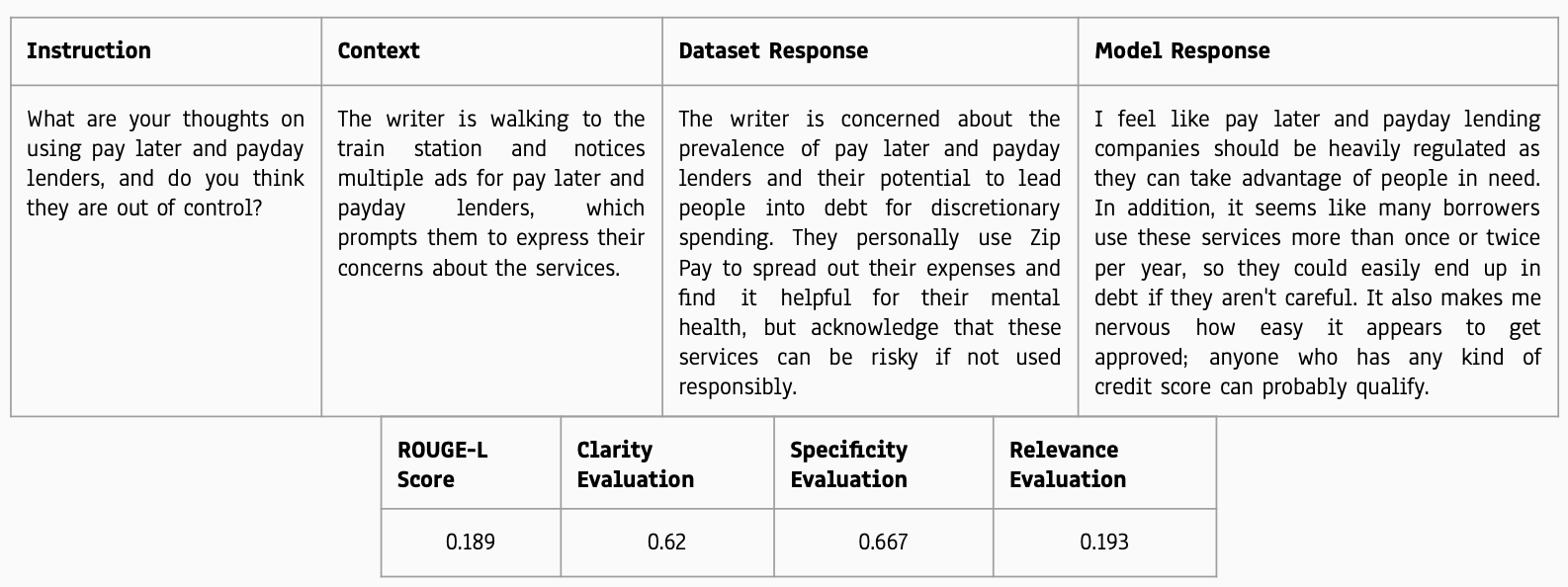


Fig 5: Case-by-case analysis of custom heuristics with context

When sufficient context is provided, the clarity score improves significantly. The text generated under these conditions is much clearer, and more concise, and demonstrates a higher degree of alignment with user expectations.

**Baseline vs Fine-tuned Model Evalution**

| Example | Instructions | Context | Baseline response | LLaMA Fine-Tuned  Response | Mistral Fine-Tuned  Response |
| --- | --- | --- | --- | --- | --- |
| 1 | When should I get a  second credit card? | Seeking advice on managing credit cards and considering when to apply for a second one. | The optimal time to consider getting a second credit card is when you have fully managed your existing credit | You can apply for your first credit card when you are 18 years old and have a steady income, such as from employment or investments. You may also be able to qualify if you're married or live with someone who has a good credit history. If you don’t meet these requirements yet, it might make sense to wait until you do before applying for another credit card. | In general, you should only get one or two credit cards at a time, and only if you can pay the bill in full each month. If you get several credit cards in a short period of time, you may not have enough available credit to keep your credit utilization ratio low, which could hurt your credit scores. |
| 2 | Tell me about the risks in money market accounts? | Exploring financial instruments and understanding potential risks associated with money market accounts. | Money market accounts carry risks such as interest rate fluctuations affecting yields, and potential penalties on frequent withdrawals. | Money Market Accounts are considered to be one of the safest investments available today, but they do have some risk factors associated with them. The first and most important thing you need to know when it comes to MMA is that there is no guarantee on your principal amount invested. Another risk factor is liquidity. | Money market accounts are vulnerable to interest rate volatility, which can alter returns, and face potential credit risks in the securities they invest in, alongside regulatory withdrawal limitations. |
| 3 | How do I find a good mutual fund to invest 5K in with a moderately high amount of risk? | NA | To find a good mutual fund for a $5,000 investment with a moderately high-risk level, research funds that match your risk tolerance and investment goals. Check past performance, fees, and fund manager history to make an informed decision. | The best way for you to invest $5000 is to take advantage of your employer's retirement plan, which will allow you to set aside money tax-free and automatically have it invested into funds based on your age and goals. If your company does not offer this benefit, then you should look at Vanguard (VG) or Fidelity (FD). These are two large discount brokerage firms that have low fees and offer great diversification by owning a variety of stocks and bonds rather than just one or two mutual funds. Both companies provide online tools where you can create an account and start investing immediately. Good luck! | For investing $5,000 in a mutual fund with moderately high risk, focus on diversified funds that invest in growth sectors or mid-cap stocks. Analyze the performance history, manager expertise, and cost structure, including expense ratios and load fees, to identify funds that offer a balanced approach to risk and potential returns. |

### **7. Discussion**

The results from FinAdvisor underscore the effectiveness of employing PEFT with QLoRA for resource-efficient fine-tuning of large language models. The tool demonstrates the capability to deliver personalized financial advice with high contextual accuracy and clarity, despite operating under computational constraints. These outcomes highlight the transformative potential of AI-driven solutions in enhancing the accessibility and usability of financial advisory services.

The project findings have tangible applications in delivering real-time financial insights, including advice on budgeting, investment strategies, and credit optimization. The seamless interaction facilitated by the Gradio-based interface significantly enhances accessibility, providing a user-friendly experience for individuals across various demographics. This integration of advanced AI with an intuitive user interface ensures FinAdvisor's value as a practical and efficient tool for modern financial management.

While the project achieved significant milestones, areas for improvement remain. As a future work, the data pipeline can optimized with a vector database to improve retrieval speeds, enhancing the system's overall response time. Additionally, refining evaluation metrics to align better with user expectations will provide more robust measures of performance. Future iterations of FinAdvisor could incorporate interactive features such as financial simulations and scenario planning, broadening the scope of applications and offering users deeper insights into their financial decisions.

#### **8. Statement of Contributions**

The development of the FinAdvisor project was a collaborative effort involving three team members, each contributing to critical aspects of the project. Their contributions are detailed below:

**Laasya Anantha Prasad** focused on dataset preprocessing and exploratory data analysis to ensure the data was clean, consistent, and suitable for training. Laasya designed and implemented the training pipeline, establishing the foundation for the fine-tuning of models and collaborating to develop the RESTful API.

**Rakshak Kunchum** led the fine-tuning process by adapting advanced models, such as LLaMA and Mistral, using techniques like Parameter-Efficient Fine-Tuning (PEFT) and Quantized LoRA (QLoRA). Rakshak also led the development of the RESTful API, enabling seamless communication between the backend models and the user interface, alongside developing a Gradio-based user interface to ensure the tool is intuitive, accessible, and user-friendly.

**Krishna Venkatesh** designed and implemented the inference pipeline to transform user queries into embeddings, retrieve relevant data, and generate personalized responses. Krishna also led the fine-tuning of the Mistral model, enhancing its adaptability and performance for specific tasks.

### **9. References**

1. Alpaca Real-Time News API. *Alpaca Market News API Documentation.* Available at: <https://docs.alpaca.markets/docs/streaming-real-time-news>
2. Gaurav Bharti. "Finance Alpaca Dataset." *Hugging Face Datasets*. Available at: <https://huggingface.co/datasets/gbharti/finance-alpaca>
3. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., & Chen, W. "*LoRA: Low-Rank Adaptation of Large Language Models.*" *arXiv preprint arXiv:2106.09685*. Available at: <https://arxiv.org/abs/2106.09685>
4. Dettmers, T., Pagnoni, A., & Zettlemoyer, L. "*QLoRA: Efficient Finetuning of Quantized LLMs*." arXiv preprint arXiv:2305.14314. Available at:<https://arxiv.org/pdf/2305.14314>
5. Weights and Biases, Inc. "Weights and Biases Documentation" Available at:<https://wandb.ai/site>
6. Gradio. "Gradio Documentation." Available at: <https://gradio.app/docs/>

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### **10. Appendix:**

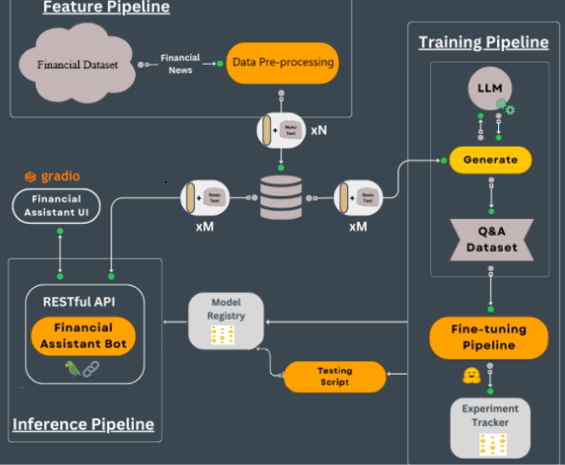


Fig 6. FinAdvisor Architecture ([LLMs kit: Build a production-ready real-time financial advisor](https://medium.com/decodingml/the-llms-kit-build-a-production-ready-real-time-financial-advisor-system-using-streaming-ffdcb2b50714))

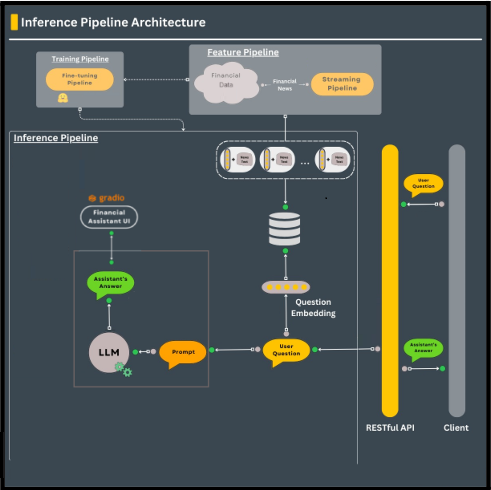


Fig 7: Inference Pipeline ([LLMs kit: Build a production-ready real-time financial advisor](https://medium.com/decodingml/the-llms-kit-build-a-production-ready-real-time-financial-advisor-system-using-streaming-ffdcb2b50714))

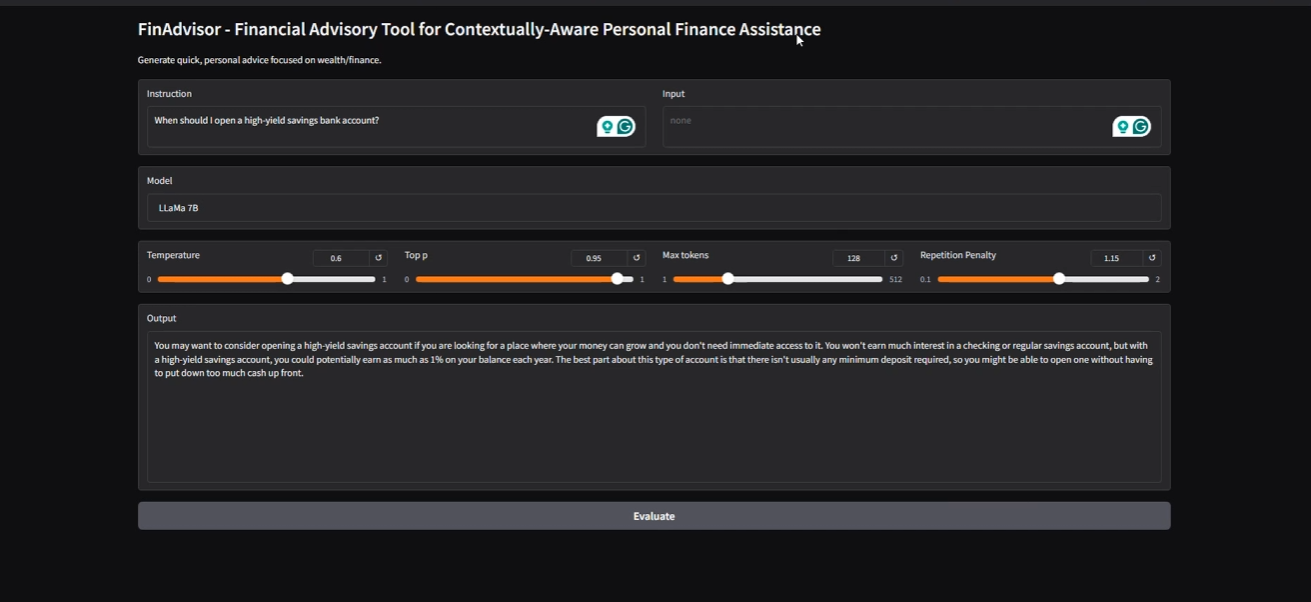


Fig 8: Gradio User Interface

### 