**P2 Preliminary Report**

### **DS5500: Data Science Capstone**

Prof. Kylie A. Bemis

### **1. Title:** FinAdvisor - Financial Advisory Tool for contextually-aware personal finance assistance.

### **2. Authors:**

* Laasya Anantha Prasad
* Krishna Venkatesh
* Rakshak Kunchum

### **3. Github:** https://github.com/sriksven/FinAdvisor\_LLM

### **4. Summary:**

Traditional financial advisory methods face significant limitations, including stouatic knowledge, generalized advice, and restricted accessibility. These constraints hinder their ability to provide timely and contextually relevant financial insights, leaving individuals without the tools necessary to make informed decisions in dynamic financial environments. The FinAdvisor project addresses these shortcomings by leveraging advanced Large Language Models (LLMs) to deliver personalized, real-time financial advice. Focusing on key areas such as budgeting, investments, and credit management, the project provides actionable, dynamic insights tailored to the complexities of modern financial needs. By integrating machine learning techniques with an intuitive user interface, FinAdvisor democratizes access to high-quality financial advisory services, empowering users to take control of their financial well-being efficiently and effectively.

The proposed solution is an AI-driven financial advisory tool powered by LLMs fine-tuned for financial contexts using diverse real-world datasets, including the Financial Alpaca Dataset and the FinTalk-19k Dataset. Unlike traditional methods that rely on static knowledge and are often limited by availability or time-consuming processes, this tool offers personalized, context-aware support 24/7. It addresses user queries dynamically by combining three key training features: instruction (user's query), context (relevant financial information), and response (tailored advice). This ensures that the guidance provided is actionable and specific to the user's needs. The result is a user-friendly web application targeted for any individual seeking quick and reliable financial advice. The FinAdvisor bridges the gap between outdated traditional tools and the demands of modern financial planning, providing convenient, efficient, and effective financial management support.

### 

### 

### **5. Phase 1 -Work Summary**

FinAdvisor's development begins with robust data preprocessing and embedding analysis, ensuring data consistency and preparing it for effective model training. The datasets were cleaned to remove extraneous white spaces, dashes, and non-ASCII characters while embedding analysis was conducted using the Sentence Transformer model (*all-MiniLM-L6-v2*). This process enabled the extraction of relevant financial context, optimizing the dataset for fine-tuning large language models.

Exploratory Data Analysis (EDA) provided critical insights into the relationships between key dataset features. It revealed that the lengths of instructions, contexts, and responses are only minimally correlated, with a Pearson correlation coefficient of -0.17. This minimal correlation indicates that the model’s response length is not overly dependent on the verbosity of user instructions or context. As a result, the model can generate advice that is clear and concise, regardless of input length. This independence enhances the user experience by ensuring that responses are succinct yet comprehensive.

The fine-tuning process employed **Low-Rank Adaptation (LoRA)** and its advanced configuration, **Quantized LoRA (QLoRA)**. LoRA is a parameter-efficient fine-tuning method that adapts language models to specialized contexts without requiring full retraining. By modifying only a small subset of parameters (e.g., q\_proj and v\_proj), LoRA significantly reduces computational and memory requirements while preserving the pre-trained model’s integrity. This approach enables efficient task-specific learning, making it particularly suitable for hardware with limited resources. QLoRA further enhances this efficiency by fine-tuning quantized (8-bit precision) versions of the model weights, balancing performance with computational efficiency.

The primary models fine-tuned in this project include **LLaMA 7B** and **Mistral 7B**. LLaMA 7B, known for its versatility and computational efficiency, has successfully completed fine-tuning, while Mistral 7B, which employs sparse attention mechanisms for high performance with reduced resource demands, is in the final stages of its fine-tuning process. Performance tracking during training and fine-tuning was managed using **Weights & Biases (wandb)**, a comprehensive experiment management tool. wandb facilitates real-time monitoring of metrics such as training loss, accuracy, and perplexity, ensuring systematic tracking of model performance.

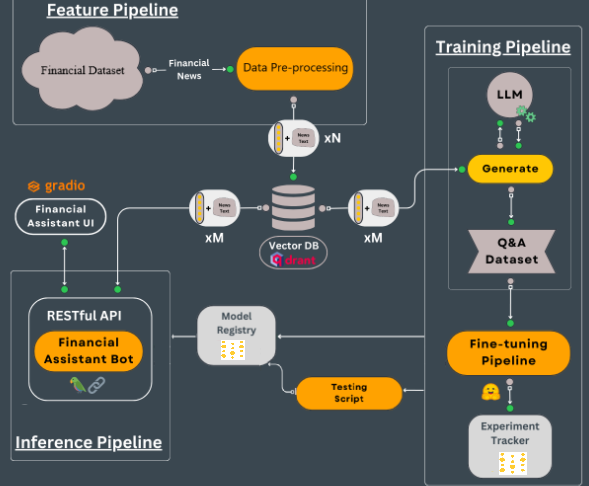
****

Fig 1: Project Architecture

**6. Phase 2 - Methods and Implementation**

The inference pipeline integrates both backend and frontend components to provide a seamless user experience. A **RESTful API** is being developed as the bridge between the fine-tuned models and the user interface. This API processes user queries, sends them to the models for inference, and returns tailored financial advice. Designed with scalability and efficiency in mind, the API is equipped to handle multiple concurrent requests without compromising performance. Additionally, it incorporates security features such as authentication and encryption to safeguard sensitive financial data, ensuring user trust and privacy.

The inference pipeline for the above-mentioned models is currently under development. This pipeline enables real-time financial advice generation by converting user queries into embeddings, retrieving relevant data from a vector database powered by Qdrant, and using fine-tuned models to generate responses.

The **user interface**, built using **Gradio**, offers an intuitive platform for users to interact with FinAdvisor. Designed for simplicity and accessibility, the interface allows users to input financial questions and receive real-time, personalized advice. The Gradio-based UI includes user-friendly features such as customizable inputs, clear error handling, and responsive design, ensuring a seamless experience across devices. Together with the RESTful API, the interface delivers a robust and efficient solution for users seeking AI-driven financial guidance. Both the API and front-end development are ongoing, with targeted completion dates aligning with the project timeline.

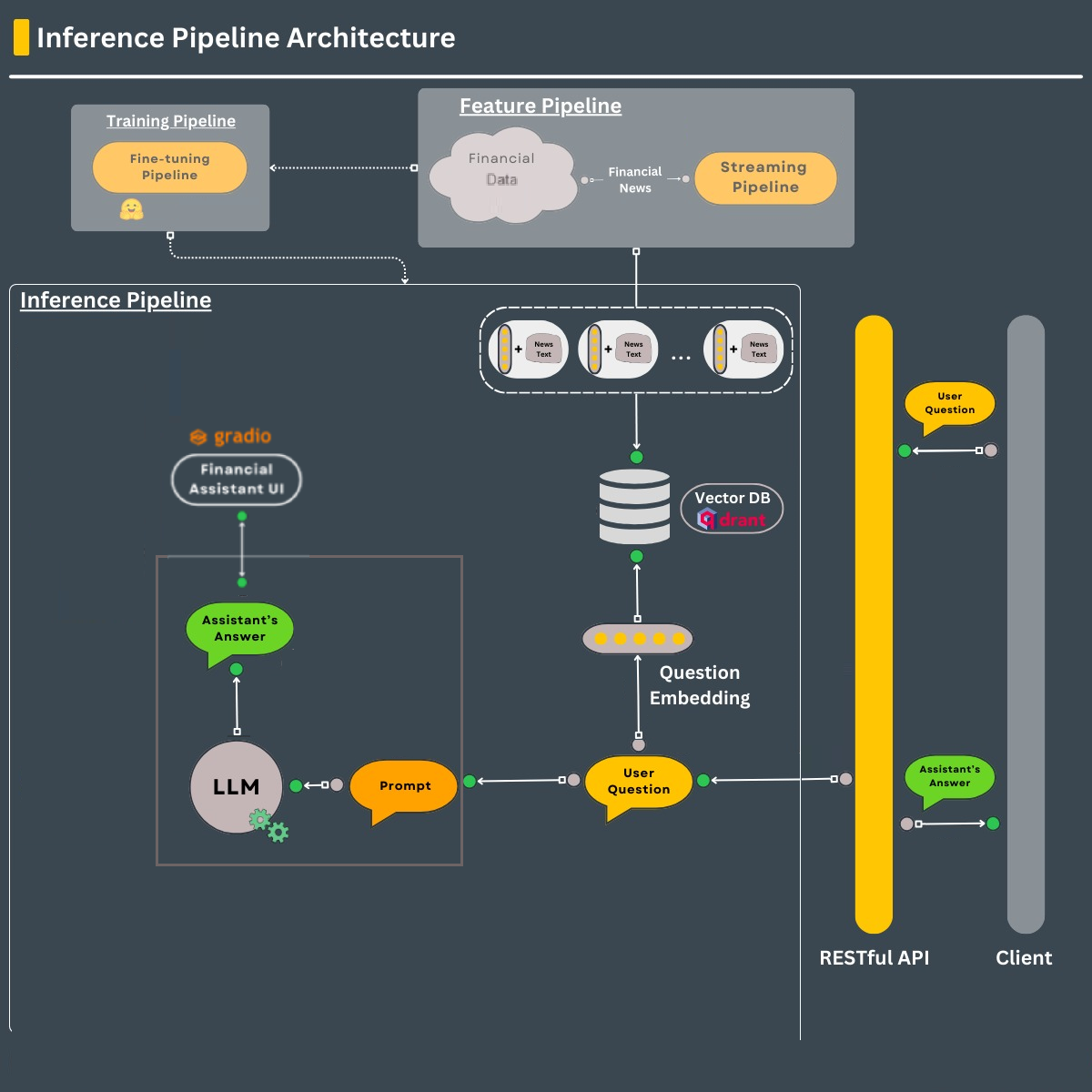


Fig 2: Inference Pipeline

### **7. 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. For the fine-tuned LLaMA model, a perplexity score of 7.12 was achieved, reflecting strong performance close to state-of-the-art benchmarks. 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. While this is a simplified metric, it is effective for identifying whether the response structure supports ease of understanding.

**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. 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.

#### **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.

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.

### **8. 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 effectiveness of the model in processing user queries and generating personalized responses was validated through a rigorous evaluation framework. While traditional 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.

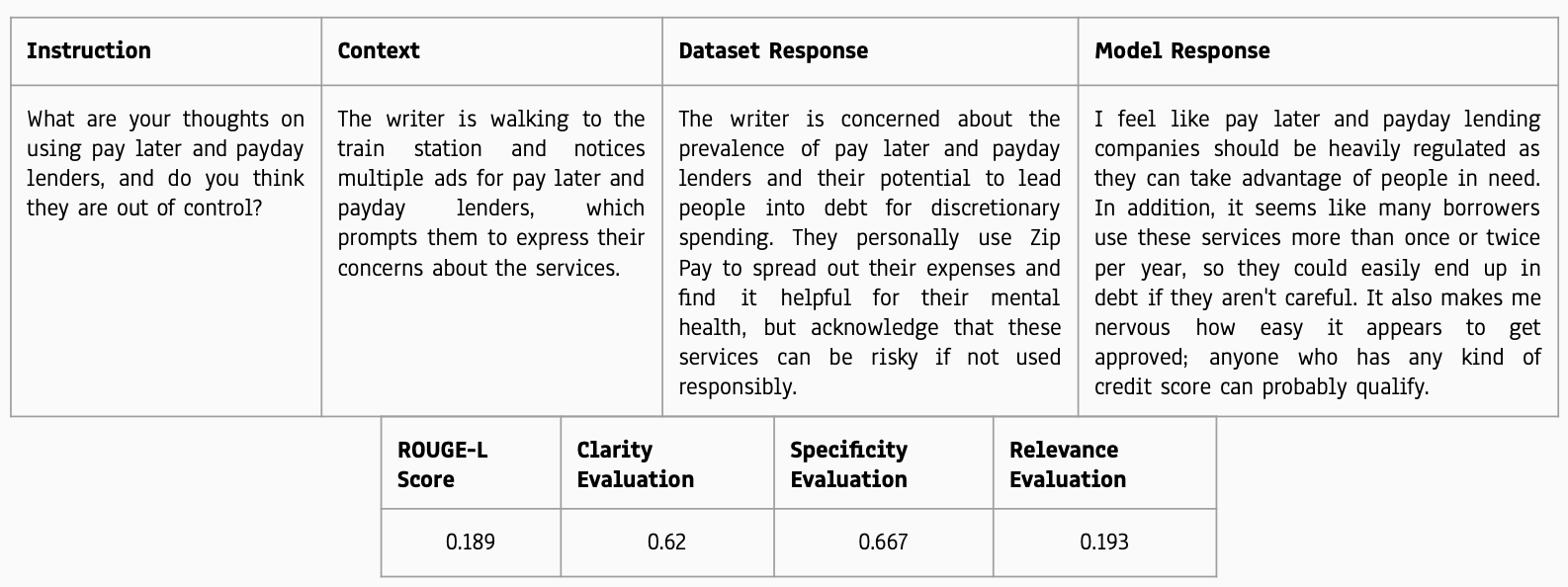


Fig 3: Case analysis of custom heuristics

### **9. 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. The vector data pipeline can be further optimized to improve retrieval speeds, enhancing the overall response time of the system. 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.

#### **10. 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:

1. **Laasya Anantha Prasad:**

Focused on dataset preprocessing and exploratory data analysis to ensure the data was clean, consistent, and suitable for training. Designed and implemented the training pipeline, establishing the foundation for the fine-tuning of models and collaborating to develop the RESTful API.

1. **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), also leading the development of the RESTful API, enabling seamless communication between the backend models and the user interface.

1. **Krishna Venkatesh:**

Designed the inference pipeline to transform user queries into embeddings, retrieve relevant data, and generate personalized responses.Currently Developing a Gradio-based user interface to ensure the tool was intuitive, accessible, and user-friendly.

### **11. 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. Izacard, G., & Grave, E. "Leveraging Passage Retrieval with Generative Models for Qdrant. "Qdrant Vector Database Documentation." Available at: <https://qdrant.tech/>Open Domain Question Answering." *arXiv preprint arXiv:2007.01282*. Available at: <https://arxiv.org/abs/2007.01282>
5. Gradio. "Gradio Documentation." Available at: https://gradio.app/docs/
6. Comet. "Comet Machine Learning Experiment Management." Available at: <https://www.comet.ml/>