Abstract

This project engages with the FinGPT-Forecaster, a state-of-the-art financial Large Language Model (LLM), to explore its efficacy in synthesizing market insights and predicting stock price movements. We fine-tuned and compared two advanced models, LLaMA-3.1 8B and DeepSeek-R1-Distill-Llama-8B, using Low-Rank Adaptation (LoRA) to enhance their performance in financial forecasting. Our analysis focused on a range of metrics including training and evaluation loss, evaluation speed, and accuracy in different financial scenarios. The findings suggest that while DeepSeek-R1 is suited for high-frequency trading due to its faster processing capabilities, LLaMA-3.1 8B offers greater accuracy and reliability for risk management and financial forecasting. The project highlights the potential for further enhancing predictive accuracy by expanding data sources and optimizing LoRA parameters, pave the way on for future advancements in financial technology using LLMs.

In the rapidly evolving field of financial technology, the integration of artificial intelligence has revolutionized how market data is analyzed and interpreted. A Columbia Menter Project offered an exceptional opportunity to engage with the FinGPT-Forecaster, a state-of-the-art financial Large Language Model (LLM) designed to synthesize market insights and predict stock price movements. This assignment tasked us with leveraging the existing capabilities of the FinGPT-Forecaster and further enhancing its predictive accuracy by fine-tuning models such as LLaMA-3.1 8B and DeepSeek-R1-Distill-Llama-8B.

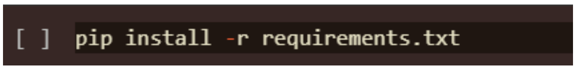
The primary objective of this study is to compare the effectiveness of the LLaMA-3.1 8B and DeepSeek-R1 models in a financial context, focusing on various performance metrics that influence their practical application in real-world scenarios. This comparative analysis aims to identify which model architecture provides the best balance between speed, accuracy, and efficiency in forecasting and risk assessment tasks.

This report is organized into several key sections:

- ✓ Setup and Execution: Describes the technical setup, including the computational environment and the data preparation process.
- ✓ Comparative analysis: Details the methods used for fine-tuning the models and the metrics for evaluating their performance.
- ✓ Impacts on financial applications: The choice between DeepSeek-R1 8B and LLaMA-3.1 8B from the comparative analysis and how could they used in different financial applications
- ✓ Conclusion and Future Work: Summarizes the key insights and explores potential directions for further enhancing the models' forecasting capabilities.

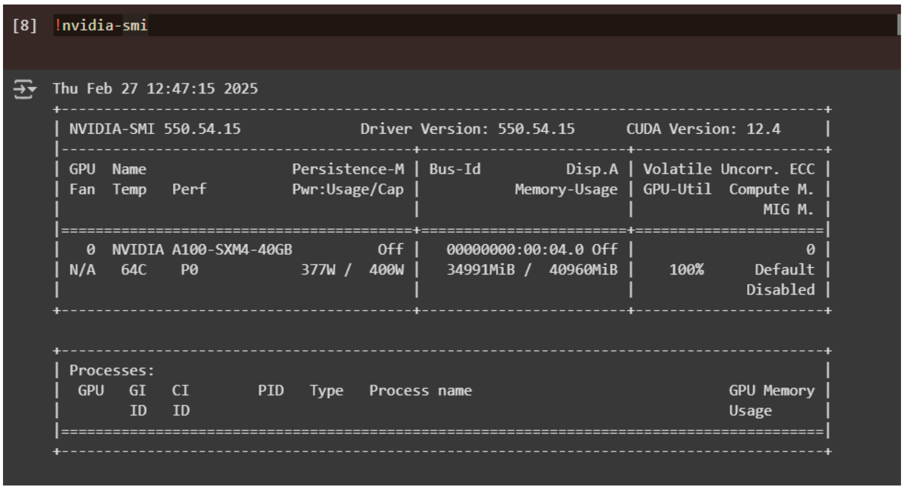
1. Setup and execution process descriptions.

   - Setup environment and GPU
     - run requirements.txt to setup running environments

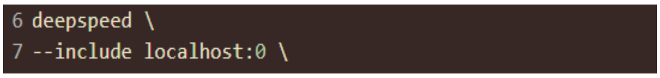


```
[ ]  pip install -r requirements.txt
```

     - setup your running GPU , I am using colab pro which provide the A100 GPU , I could find GPU position on position 0 . according to GPU position update train.sh file deepseed setting.



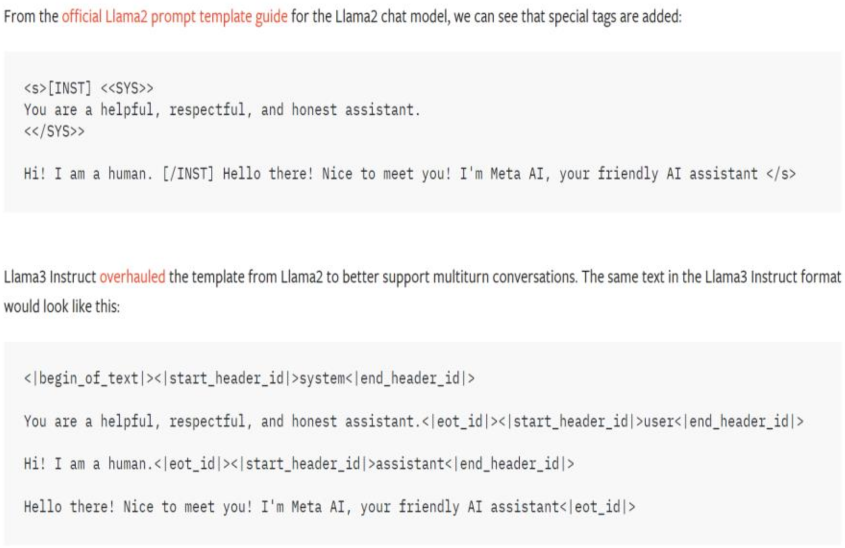
```
[8] !nvidia-smi

Thu Feb 27 12:47:15 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA A100-SXM4-40GB          Off |   00000000:00:04.0 Off |                    0 |
| N/A   64C    P0            377W /  400W |   34991MiB /  40960MiB |    100%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
+-----------------------------------------------------------------------------------------+
```



```
deepspeed \
--include localhost:0 \
```

- check your config.json file to ensure if you have ZeroOneAdam set , you should keep zero_optimization as stage 0 , I met issues where I didn’t setup up it correctly.

- Setup correct dataset chat template as we will use llama3 8B model and DeepSeekR1 model which have different chat template requirements ,
  - Llama2 & Llama3 different you could find from : https://pytorch.org/torchtune/0.3/tutorials/chat.html



From the official Llama2 prompt template guide for the Llama2 chat model, we can see that special tags are added:

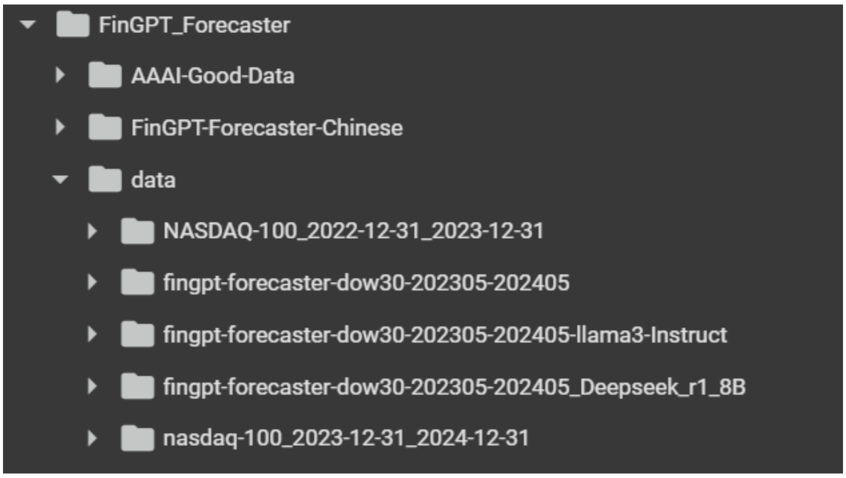
```
<s>[INST] <<SYS>>
You are a helpful, respectful, and honest assistant.
<</SYS>>

Hi! I am a human. [/INST] Hello there! Nice to meet you! I'm Meta AI, your friendly AI assistant </s>
```

Llama3 Instruct overhauled the template from Llama2 to better support multiturn conversations. The same text in the Llama3 Instruct format would look like this:

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>

You are a helpful, respectful, and honest assistant.<|eot_id|><|start_header_id|>user<|end_header_id|>

Hi! I am a human.<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Hello there! Nice to meet you! I'm Meta AI, your friendly AI assistant<|eot_id|>
```

  - For deepseek don’t have official documentation talk about the chat template, but it was mentioned that DeepSeek uses a non-display tagging system, through a modular typesetting format (similar to markdown), using text to define roles, conversation levels, contextual connections, etc. but it mentions.
  - Prepared chat template transformation script

(llama3_dsr18b_datasetpreparation_dow30.py), which could use transfer llama2 dataset to llama3-8B-Instruct and DeepSeek R1-8B model and save the dataset in local ./data/ directory,



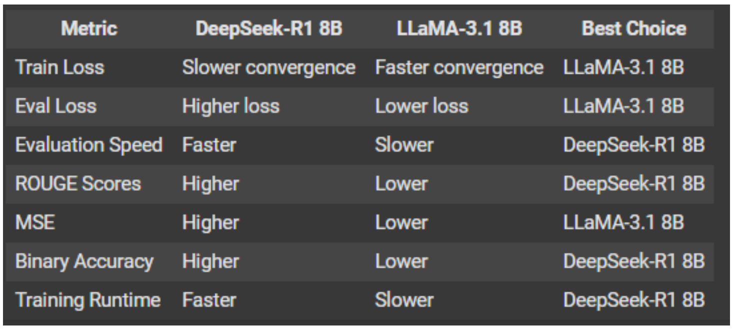
```
FinGPT_Forecaster
    AAAI-Good-Data
    FinGPT-Forecaster-Chinese
    data
        NASDAQ-100_2022-12-31_2023-12-31
        fingpt-forecaster-dow30-202305-202405
        fingpt-forecaster-dow30-202305-202405-llama3-Instruct
        fingpt-forecaster-dow30-202305-202405_Deepseek_r1_8B
        nasdaq-100_2023-12-31_2024-12-31
```

- Run train.sh to start training

2. To comparative analysis of the two models. We take six metrics to compare two models .

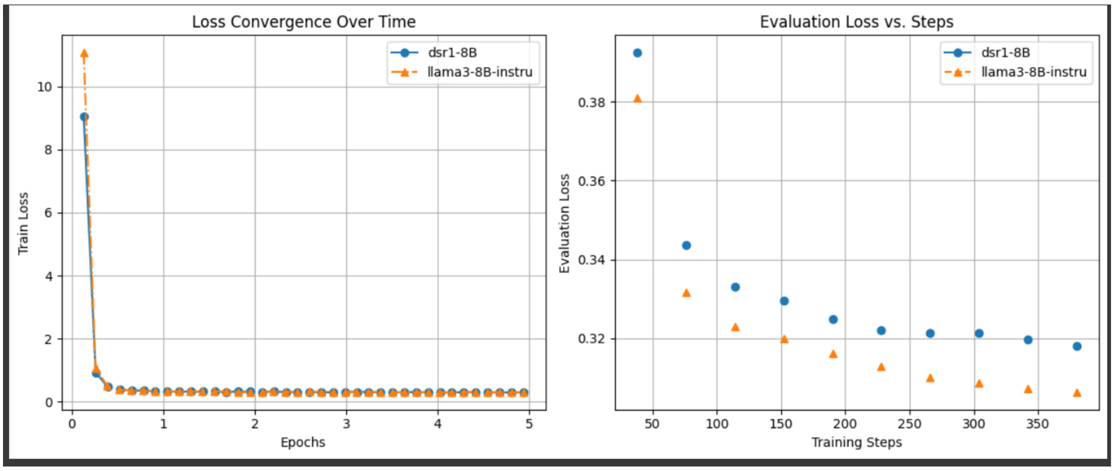
   - Training and Evaluation Loss
   - Evaluation Speed (Samples per Second)
   - ROUGE Scores (1, 2, L)
   - Mean Squared Error (MSE)
   - Binary Accuracy
   - Training Runtime & Efficiency

   After evaluation , we could see the below matrix between two models :

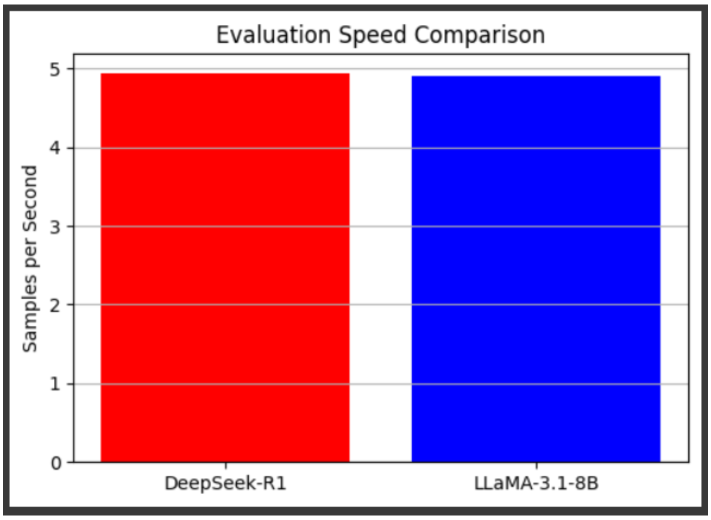


Metric	DeepSeek-R1 8B	LLaMA-3.1 8B	Best Choice
Train Loss	Slower convergence	Faster convergence	LLaMA-3.1 8B
Eval Loss	Higher loss	Lower loss	LLaMA-3.1 8B
Evaluation Speed	Faster	Slower	DeepSeek-R1 8B
ROUGE Scores	Higher	Lower	DeepSeek-R1 8B
MSE	Higher	Lower	LLaMA-3.1 8B
Binary Accuracy	Higher	Lower	DeepSeek-R1 8B
Training Runtime	Faster	Slower	DeepSeek-R1 8B

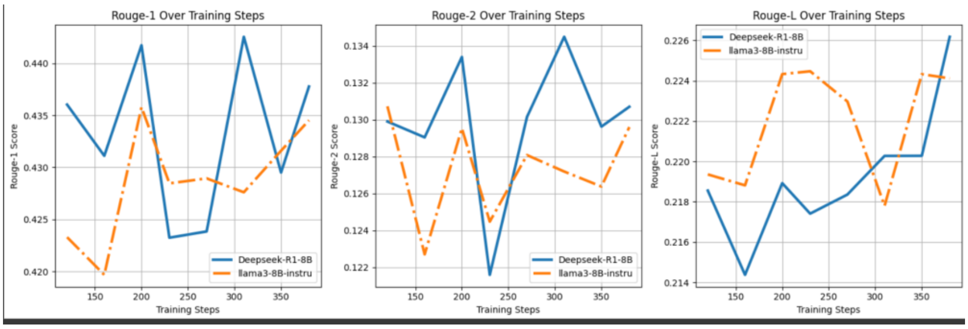
- The **Training loss convergence overtime** and **Evaluation loss vs Steps** is as below which shows : LLaMA-3.1 8B appears to learn a little bit **faster** and may require **fewer training epochs** to reach optimal performance. **DeepSeek-R1 8B has a slightly higher evaluation loss**, meaning it might be more sensitive to overfitting, So adjusting learning rates, batch sizes, or regularization parameters could be tested in future to help in optimizing "drs1-8B" to lower its evaluation loss.



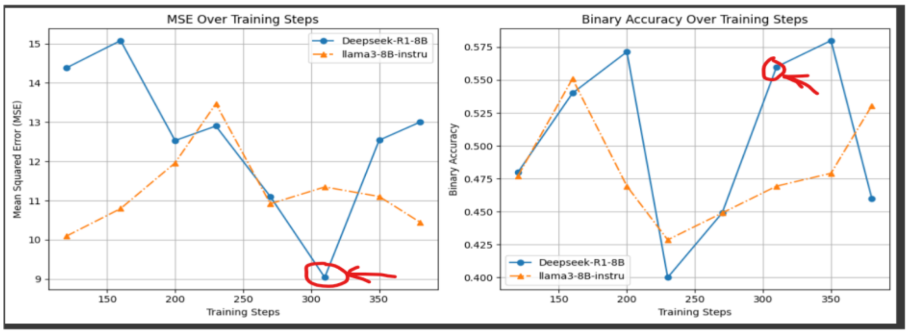
- Evaluation Speed (Samples per Second): **DeepSeek-R1 8B processes more samples per second** compared to LLaMA-3.1 8B. It shows for applications requiring **real-time inference**, **DeepSeek-R1 8B is preferable**.



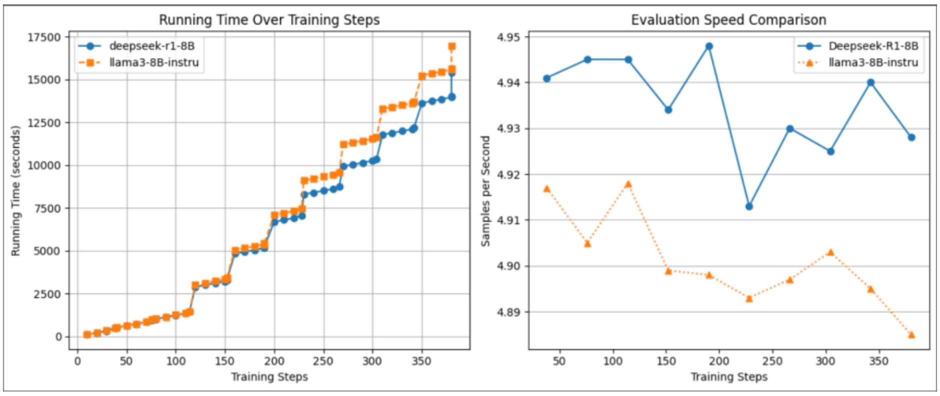
- ROUGE Scores (Text Quality) : **DeepSeek-R1 8B slightly outperforms LLaMA-3.1 8B** on **ROUGE-1, ROUGE-2, and ROUGE-L** which means For NLP tasks focused on text summarization or quality, **DeepSeek-R1 8B** is the better choice



- Binary Accuracy : Select Earlier checkpoint(red arrow) , MSE is at a local minimum and Binary Accuracy is relatively high . it shows DeepSeek-R1 8B achieves higher binary accuracy during training, which means for classification tasks, DeepSeek-R1 8B is the better choice.



- Training Runtime and Efficiency: **DeepSeek-R1 8B trains faster per step** than LLaMA-3.1 8B.



3. Impacts on financial applications from LLaMA-3.1 8B and DeepSeek-R1 8B：

   A. Risk Management and Forecasting: The lower MSE of LLaMA-3.1 8B suggests it could be more reliable for predicting financial quantities where precision is paramount, such as in risk assessment and financial forecasting. Faster convergence and lower evaluation loss imply that LLaMA-3.1 8B can more quickly adapt to new data, important in the volatile financial market where patterns can change unexpectedly.

   B. Real-time Decision Making: The faster evaluation speed of DeepSeek-R1 8B might be preferable in trading algorithms or other financial applications where decisions must be made in milliseconds, such as high-frequency trading.

   C. Fraud Detection: The higher binary accuracy of DeepSeek-R1 8B suggests it could be more effective in identifying fraudulent transactions, which are typically binary (fraud/no-fraud) classification tasks.

   D. Regulatory Compliance and Reporting: Higher ROUGE scores for DeepSeek-R1 8B indicate it could better handle tasks like generating compliance reports or summarizing financial documents where capturing the essence of texts accurately is crucial.

E. Scalability and Maintenance: The faster training runtime of DeepSeek-R1 8B supports environments where models need regular updates with new data, such as models used in dynamic markets or for regulatory compliance that frequently changes.

As summarization: the choice between DeepSeek-R1 8B and LLaMA-3.1 8B should be guided by the specific requirements of the financial application:

- LLaMA-3.1 8B is suited for applications where high accuracy, low error rates, and predictive reliability outweigh the need for speed, such as in risk management and detailed financial forecasting.
- DeepSeek-R1 8B would be advantageous in scenarios requiring fast response times and high throughput, like high-frequency trading, or where binary classification tasks such as fraud detection are prevalent.

4. Future Model Improvements :

Due to time constraints, this project represents an **initial exploration** into fine-tuning **LLaMA-3.1 8B** and **DeepSeek-R1 8B** models for financial forecasting. While we gained valuable insights into the **performance trade-offs** between these models, further research is required to fully optimize their predictive capabilities. Below are key areas for future improvements:

A. LoRA Parameter Optimization: Fine-tuning **Low-Rank Adaptation (LoRA) parameters** could significantly enhance model efficiency and performance. Future work should focus on:

- **Hyperparameter Tuning:** Systematic adjustment of LoRA parameters such as **rank** and **learning rate** using **grid search** or **Bayesian optimization** to find the optimal balance between **model complexity and predictive accuracy**.

B. Data Enrichment and Expansion： Expanding the dataset can **improve model generalization** and enable better performance across diverse financial conditions. Future efforts may include:

- **Longer Time Frames:** Incorporating extended historical financial data to capture broader market trends.
- **Additional Data Sources:** Integrating **macroeconomic indicators, alternative financial signals**, or **real-time market data** to enhance model

robustness.

C. Advanced Feature Engineering: Developing **more sophisticated features** from existing data could further refine model predictions. Key areas of focus include:

- **Technical Indicators:** Implementing commonly used **stock market indicators**, such as:
  - **Moving Averages** (SMA, EMA)
  - **Relative Strength Index (RSI)**
  - **Moving Average Convergence Divergence (MACD)**
    These indicators could enhance the model's ability to capture **market momentum and trend reversals**.