Using LLaMa3 to Project Stock Direction

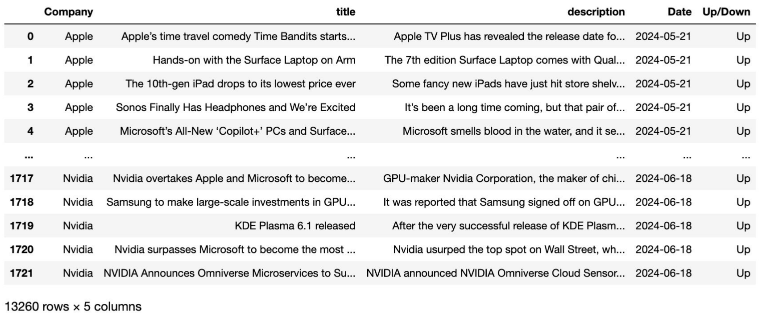
Shreyas Anil, Ryan Bernstein, Manas Nambiar, Justin Yang

Problem Statement

- Financial markets are highly influenced by news, with significant events causing rapid changes in stock prices
- The vast amount of daily news articles makes it challenging for investors to manually analyze and assess their impact on stock prices
- **Goal**: Streamline this process by using an API to pull news articles and train an LLM to predict whether a company's stock would go Up or Down based on that days news articles title and description

Our Data and Pipeline

- NewsAPI to get Articles
- Alpha Vantage API to get Stock Data
- GitHub Actions for daily updates on a Scheduler
- We looked at 7 different companies, and recorded whether their stock went up or down for each trading day of the last month. We also scraped articles written about the company during that month. We paired each article title and description with its associated date (defined as between 1:00 PM PST on the previous trading day and 1:00 PM PST on the current trading day).
- Our goal is to use the articles to predict Up/Down

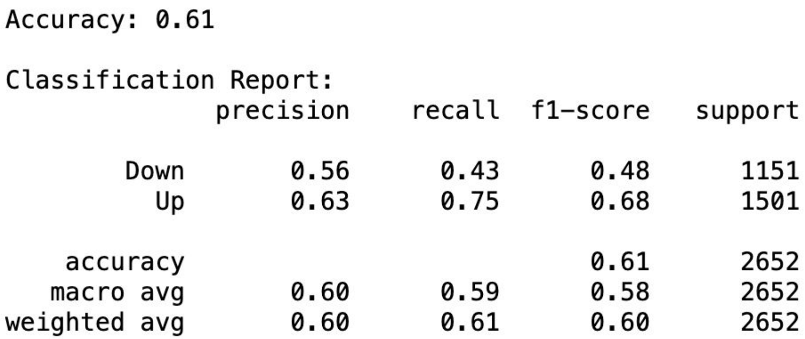


	Company	title	description	Date	Up/Down
0	Apple	Apple's time travel comedy Time Bandits starts...	Apple TV Plus has revealed the release date fo...	2024-05-21	Up
1	Apple	Hands-on with the Surface Laptop on Arm	The 7th edition Surface Laptop comes with Qual...	2024-05-21	Up
2	Apple	The 10th-gen iPad drops to its lowest price ever	Some fancy new iPads have just hit store shelv...	2024-05-21	Up
3	Apple	Sonos Finally Has Headphones and We're Excited	It's been a long time coming, but that pair of...	2024-05-21	Up
4	Apple	Microsoft's All-New 'Copilot+' PCs and Surface...	Microsoft smells blood in the water, and it se...	2024-05-21	Up
...	...	...	...	...	...
1717	Nvidia	Nvidia overtakes Apple and Microsoft to become...	GPU-maker Nvidia Corporation, the maker of chi...	2024-06-18	Up
1718	Nvidia	Samsung to make large-scale investments in GPU...	It was reported that Samsung signed off on GPU...	2024-06-18	Up
1719	Nvidia	KDE Plasma 6.1 released	After the very successful release of KDE Plasm...	2024-06-18	Up
1720	Nvidia	Nvidia surpasses Microsoft to become the most ...	Nvidia usurped the top spot on Wall Street, wh...	2024-06-18	Up
1721	Nvidia	NVIDIA Announces Omniverse Microservices to Su...	NVIDIA announced NVIDIA Omniverse Cloud Sensor...	2024-06-18	Up

13260 rows × 5 columns

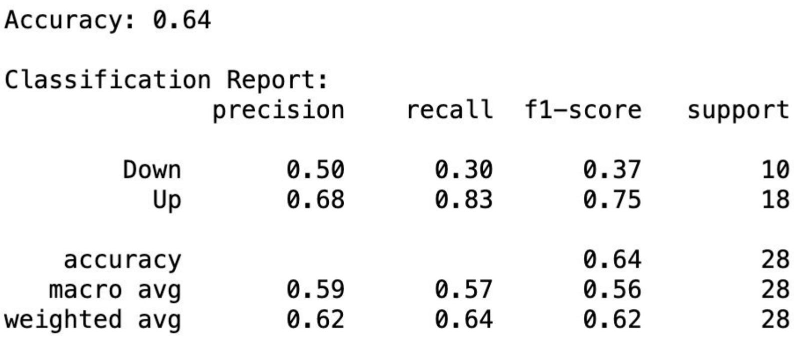
Baseline Results (from Naive Bayes and TFIDF)

Trained/Tested on Each Individual Article (13260 data points)



```
Accuracy: 0.61

Classification Report:
              precision    recall  f1-score   support

        Down       0.56      0.43      0.48      1151
          Up       0.63      0.75      0.68      1501

    accuracy                           0.61      2652
   macro avg       0.60      0.59      0.58      2652
weighted avg       0.60      0.61      0.60      2652
```

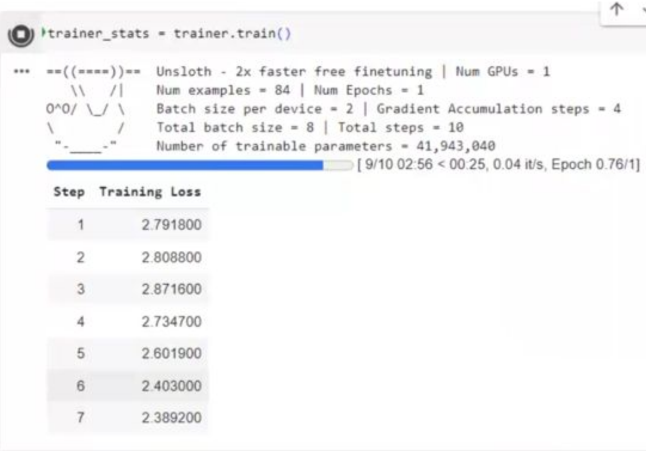
Trained/Tested on Articles Aggregated by Company and Date (140 Data Points)



```
Accuracy: 0.64

Classification Report:
              precision    recall  f1-score   support

        Down       0.50      0.30      0.37        10
          Up       0.68      0.83      0.75        18

    accuracy                           0.64        28
   macro avg       0.59      0.57      0.56        28
weighted avg       0.62      0.64      0.62        28
```

LLM Pipeline

- Base Model: LLaMa-3 8b Instruct (4-bit quantized)
- Prompt Format:
  - System Prompt: “You are a helpful AI assistant for stock market tips and recommendations”
  - Provide 15 news blurbs
  - “Based on your understanding of the news, determine whether {company name}'s stock will go up or down by open tomorrow. Please respond with either (A) - UP or (B) - DOWN.<|eot_id|>”
- Fine-tuning:
  - Used Unsloth to fine-tune on single T4 GPU on Colab
  - Trained for 4 epochs
  - Default hyperparameters

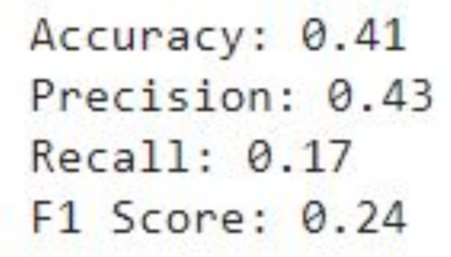


```
trainer_stats = trainer.train()
```

```
==((====))==  Unsloth - 2x faster free finetuning | Num GPUs = 1
   \\   /|    Num examples = 84 | Num Epochs = 1
O^O/ \_/ \    Batch size per device = 2 | Gradient Accumulation steps = 4
\        /    Total batch size = 8 | Total steps = 10
 "-____-"     Number of trainable parameters = 41,943,040
                                          [ 9/10 02:56 < 00:25, 0.04 it/s, Epoch 0.76/1]
Step  Training Loss
   1       2.791800
   2       2.808800
   3       2.871600
   4       2.734700
   5       2.601900
   6       2.403000
   7       2.389200
```

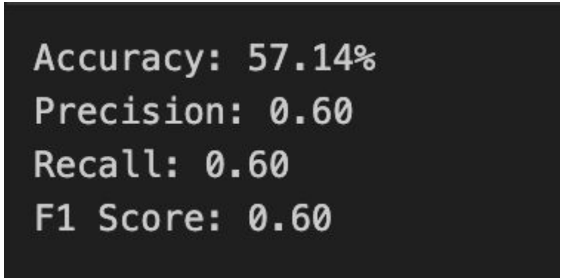
Results

Initial Model (untuned)



```
Accuracy: 0.41
Precision: 0.43
Recall: 0.17
F1 Score: 0.24
```

Fine Tuned Model



```
Accuracy: 57.14%
Precision: 0.60
Recall: 0.60
F1 Score: 0.60
```

Conclusion and Next Steps

- Fine Tuning makes a big difference with regards to this task
- Our current fine-tuned doesn’t perform significantly better than our baseline Naive Bayes/TFIDF model, so we still have room to improve
- Come up with a better strategy to filter the article that we use as inputs
- Use Ray to hyperparameter tune
- Modify our Github Actions scheduler to incorporate scheduled updates to our model
- Track model versions with Weights and Biases

Enjoy Your Summer!