Economics with LLMs: Measuring, Forecasting, and Simulating Economic Behavior

Day #2: Natural Language Understanding and Measurement

Sophia Kazinnik

Digital Economy Lab
Stanford University

August 18, 2026

Today's Plan

- **First block**: Measuring latent concepts in text
- **Second block**: Local LLMs, explainability, and econometric validation
- **Main point**: The goal is to produce a valid, interpretable regressor or label that allows for meaningful causal or structural inference

Class Outline

How can LLMs be used to understand and process economic text:

- Central bank communication:
  - *Can ChatGPT Decipher Fedspeak?* with Anne Hansen
- Financial communication:
  - *Evaluating Local Language Models: An Application to Financial Earnings Calls* with Tom Cook, Anne Hansen, and Peter McAdam

Class Outline

How can LLMs be used to understand and process economic text:

- Central bank communication:
  - *Can ChatGPT Decipher Fedspeak?* with Anne Hansen
- Financial communication:
  - *Evaluating Local Language Models: An Application to Financial Earnings Calls* with Tom Cook, Anne Hansen, and Peter McAdam
- Model explainability: available methods
- Measurement validation
- Example walkthrough

What Are We Measuring?

Economists care about things that cannot be observe directly..

We observe:

- words
- choices/actions
- prices

What Are We Measuring?

Economists care about things that cannot be observe directly..

We observe:

- words
- choices/actions
- prices

But we often want to measure:

- beliefs
- expectations
- sentiment
- narratives

From Observable Data to Latent Constructs

Economists often want to measure concepts that cannot be observed directly.

Observables $\longrightarrow$ **Measurement** $\longrightarrow$ **Latent construct**

- What exactly are we trying to measure?
- What observable evidence should reveal the construct?
- Apply a consistent procedure to turn evidence into a score or classification.
- Would we obtain similar measurements across prompts, models, or repeated runs?

Example: Measuring Central Bank Stance

"Hawkishness" (a policy stance focused on fighting inflation) is not directly observed, it's a judgment we infer from language.

Research task:
Turn central bank communication into a measure of policy stance.

- Define what counts as hawkish or dovish
- Ask the model to apply that definition consistently
- Validate the measure against other evidence

Validation could include:

- human expert labels
- market reactions
- comparison with existing measures

Example: Central Bank Communication

Can ChatGPT Decipher Fedspeak?

Anne Lundgaard Hansen and Sophia Kazinnik*

April 10, 2024

Abstract

Abstract This paper examines the ability of large language models (LLMs) to decipher Fedspeak, the technical language used by the Federal Reserve to communicate on policy decisions. We evaluate the ability of the GPT-3.5 and GPT-4 models to correctly classify the policy stance of FOMC announcements relative to human assessment. We find that these models outperform traditional methods in classification accuracy and provide justifications akin to human rationale. Additionally, we show that the GPT-4 model can successfully perform the elaborate and non-trivial task of identifying macroeconomic shocks using the narrative approach of Romer and Romer (1989, 2023). Finally, we show preliminary evidence that market reactions to FOMC announcements have intensified following the introduction of the GPT-4 model.

Keywords: Large Language Models (LLMs), Artificial Intelligence (AI), Financial Markets, Central Bank Communication.

JEL Codes: E52, E58, C88.

The Question

Back in February 2023, we asked ourselves the following question:

Can ChatGPT decipher Fedspeak?

- Perhaps..?
  - Architecture + training data + billions of parameters $\rightarrow$ LLMs are really good at natural language tasks
- Perhaps not..
  - "Off the shelf" model not specifically geared towards financial text
  - Fedspeak is known to be technical and often convoluted
  - Rich in signals that go beyond policy stance

The Question

How does one answer this question?

- If "decipher" == classification: need a benchmark
- Beyond that, more difficult conceptually..

Our Strategy

(1) Classification:

- Create a benchmark dataset of manually classified FOMC announcement sentences
- Use GPT models to classify the policy stance of these sentences; compare performance to the human benchmark

(2) Reasoning:

- (Qualitatively) evaluate the ability of GPT models to explain reasoning behind the chosen classifications

(3) Narrative approach:

- Replicate the narrative approach of Romer & Romer (1989) for monetary policy shock identification

These exercises show that GPT models do "understand" Fedspeak

Implications?

Implications?

"Since I've become a central banker, I've learned to mumble with great incoherence. If I seem unduly clear to you, you must have misunderstood what I said."

— Alan Greenspan

Implications

- We are in an era where CBs are focused on making their communication more accessible
- At the same time, even with these efforts there is a communication gap (e.g., Blinder, Ehrmann, de Haan, and Jansen 2022)

    1. LLMs could reduce this gap for the general public (e.g., tailored communication)
    2. LLMs could also reduce the knowledge gap between large financial institutions and smaller entities

- More effective monetary policy transmission?
- A lot of unanswered questions...

Classification of Policy Stance

Data

- FOMC statements published between 2010 and 2020
  - Statements are released on FOMC meeting days, 8 times per year
  - Summarize the FOMC's view of the economy and policy deliberations
- We break down these statements into individual sentences
- Benchmark:
  - A group of RAs manually classified sentence sub-sample: "dovish", "mostly dovish", "neutral", "mostly hawkish", "hawkish"
  - To avoid bias, each sentence was independently reviewed by 3 RAs
  - "Truth" = human consensus
  - Manual annotation is costly; we focus on 500 randomly selected sentences

Labels: Policy Stance Categories

Classification	Value	Definition
Dovish	-1	Strongly expresses a belief that the economy may be growing too slowly and may need stimulus through monetary policy.
Mostly dovish	-0.5	Overall message expresses a belief that the economy may be growing too slowly and may need stimulus through monetary policy.
Neutral	0	Expresses neither a hawkish nor dovish view and is mostly objective.
Mostly hawkish	0.5	Overall message expresses a belief that the economy is growing too quickly and may need to be slowed down through monetary policy.
Hawkish	1	Strongly expresses a belief that the economy is growing too quickly and may need to be slowed down through monetary policy.

Summary Statistics

- The mode classification is "neutral"
- The sample is skewed towards "dovish" sentences
- Average disagreement is symmetrically U-shaped with less disagreement about "neutral" sentences and more about "dovish" and "hawkish" sentences

	Total	Dovish	Mostly Dovish	Neutral	Mostly Hawkish	Hawkish
Count	500	104	144	191	47	14
Avg. Disagreement	0.47	0.67	0.52	0.31	0.51	0.67
N (>1 step)	264	104	60	67	19	14
N (>2 steps)	49	0	21	22	6	0

Notes: Average disagreement is calculated as the average difference between the classifications assigned by the 3 reviewers using the numerical value of each classification. N (>1 step) and N (>2 steps) are the number of sentences for which maximal disagreement exceed 1 and 2 steps.

Method

- We use the GPT-3.5 and GPT-4 models implemented zero-shot to label the policy stance of each sentences:
  - GPT-3.5: TEXT-DAVINCI-003
  - GPT-4: GPT-4-0613
  - Fine-tuning (400/100 training/test data split) with GPT-3.5
- Temperature is set to 0 (minimal randomness)
- Performance = ability to replicate human labels. Measured using error statistics and confusion matrix metrics
- We benchmark the performance against SOTA and traditional NLP methods

Prompt?

For measurement, the prompt usually does three things:

- Gives the model the text to score ({excerpt})
- Defines the construct we want to measure
- Specifies the allowed output

Example

```
The following is an excerpt from a Federal Reserve speech.

"{excerpt}"

What is the monetary policy stance of this excerpt?
The options are:  hawkish, dovish, neutral.
Answer with one of these labels.
```

A prompt is a measurement instrument.

API Call?

An **API call** is how the prompt reaches the model.

$$\text{Prompt} + \text{settings} \rightarrow \text{Model provider} \rightarrow \text{Model output}$$

For measurement, the settings matter:

- Which model produced the answer?
- How much randomness is allowed?
- What kind of output allowed: free text, label, JSON, number, etc.
- What documents or prior text did the model see?

In an LLM measurement task, the prompt is only part of the method.
The model, settings, context, and output rule are part of the method too.

Other NLP Methods

- BERT (Bidirectional Encoder Representations from Transformers)
  - Transformer language model (predicts missing tokens, doesnt generate)
  - Considered state-of-the-art until LLMs and widely applied (see, for example, Bertsch et al. 2024 and Huang et al. 2022 for applications in finance)
- Dictionary-based methods:
  - Dictionary-based methods use pre-defined lexicons containing labeled words or phrases
  - Popular for their simplicity and transparency
  - Performance is limited by their coverage and they struggle with nuances and broader context of the language

Dictionary	# words
Loughran & McDonalad 2011 dictionary	2700
Henry 2008 financial dictionary	190
NRC Word-Emotion Association Lexicon 2015	11,251

Performance Measures for Multi-Class Problems

- Numeric Error Metrics:
  - **MAE (Mean Absolute Error)**: Average absolute difference between predicted and true values.
  - **RMSE (Root Mean Squared Error)**: Penalizes larger errors more than MAE.
- **Accuracy** (less informative for cases with multiple classes):
  - Proportion of total correct predictions.
  - *Warning:* Can be misleading if some classes dominate (class imbalance).
- **Balanced Accuracy** (better for uneven class sizes):
  - Average of the recall for each class.
  - *More robust than accuracy when class sizes differ.*

Performance Measures for Multi-Class Problems

- **F1 Score** (key metric for each class):
  - Combines **precision** and **recall** into one score.
  - Can be computed:
    - **Per class** – helpful to see how well each class is predicted.
    - **Macro average** – average F1 across classes (treats all classes equally).
    - **Weighted average** – averages F1 scores weighted by support (more stable with imbalanced data).
  - **Precision** = # correct positive predictions / # total predicted positives.
    Recall = # correct positive predictions / # actual positives.
  - **Type I error** (false positive) $\Rightarrow$ lowers precision
  - **Type II error** (false negative) $\Rightarrow$ lowers recall

Tip: For 5-class classification, use F1 (macro/weighted) and balanced accuracy over simple accuracy.

F1 Score: Balancing Precision and Recall

Definition: The F1 Score is the harmonic mean of precision and recall.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

- **Precision** $= \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$
- **Recall** $= \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$

Why use F1?

- Useful when classes are imbalanced or errors have different costs.
- More informative than accuracy in multi-class settings (e.g., 5-class classification).
- Low precision (Type I error) or low recall (Type II error) will both reduce F1.

Zero-Shot Performance Metrics

	GPT-4	GPT-3.5	BERT	LM	Henry	NRC
MAE	**0.31**	0.41	0.66	0.62	0.55	0.81
RMSE	**0.48**	0.58	0.84	0.80	0.75	0.96
Accuracy	**0.52**	0.37	0.25	0.28	0.35	0.11
F1 Score						
Dovish	0.42	**0.49**	0.31	0.07	0.17	0.04
Mostly dovish	0.38	**0.43**	0.33	0.23	0.04	0.17
Neutral	**0.63**	0.15	0.13	0.48	0.57	0.14
Mostly hawkish	**0.44**	0.36	NA	0.15	0.07	0.11
Hawkish	**0.47**	0.10	0.07	NA	0.08	0.03
Balanced Accuracy						
Dovish	0.63	**0.71**	0.48	0.51	0.53	0.51
Mostly dovish	**0.60**	0.56	0.56	0.53	0.50	0.51
Neutral	**0.66**	0.54	0.51	0.59	0.59	0.45
Mostly hawkish	**0.67**	**0.67**	0.50	0.49	0.50	0.42
Hawkish	**0.81**	0.53	0.52	0.47	0.56	0.45

⏩ Performance metrics ⏩ Fine-tuned learning

A Note on Fine-tuning

- **Goal:** Adapt a general model to domain-specific language and decision rules; increase label consistency and reduce prompt sensitivity.
- **Data:** 500 human-labeled sentences $\rightarrow$ 400 train / 100 test *(stratified by label; duplicates removed)*.
- **Setup:**
  - Instruction-style objective: "Classify the sentence as exactly one of `{Dovish, Mostly Dovish, Neutral, Mostly Hawkish, Hawkish}`; return only the label."
  - Same instruction template for training and evaluation; no few-shot exemplars.
  - Temperature $= 0$; optional stop sequences to enforce single-label outputs.
- **Note:** This sample is too small!

Fine-Tuned Performance Metrics

	GPT-3 (fine-tuned)	GPT-3 (zero-shot)	BERT	LM	Henry	NRC
MAE	**0.23**	0.40	0.60	0.58	0.54	0.85
RMSE	**0.40**	0.57	0.77	0.79	0.71	0.98
Accuracy	**0.61**	0.41	0.28	0.33	0.31	0.10
F1 score						
Dovish	**0.77**	0.48	0.34	0.07	0.06	0.07
Mostly dovish	**0.53**	0.45	0.31	0.34	0.07	0.26
Neutral	**0.66**	0.24	0.18	0.58	0.52	0.04
Mostly hawkish	0.22	**0.50**	NA	0.12	NA	0.11
Hawkish	**0.80**	NA	NA	NA	NA	NA
Balanced Accuracy						
Dovish	**0.83**	0.65	0.45	0.52	0.47	0.51
Mostly dovish	**0.67**	0.59	0.54	0.57	0.51	0.55
Neutral	**0.73**	0.57	0.53	0.67	0.52	0.40
Mostly hawkish	0.61	**0.80**	0.50	0.54	0.47	0.49
Hawkish	**0.99**	0.49	0.49	0.48	0.45	0.39

Note: For each metric, the best performing model is boldfaced. Performance is tested on test sample of 100 sentences only.

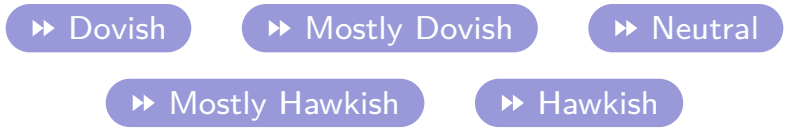
Go Back

Reasoning

Can ChatGPT Reason?

- Mere classification aside, GPT models have the ability to explain why a certain sentence was labeled in a certain way
- We test this capability in a case study:
  - We ask both ChatGPT and a human research assistant, Bryson, to classify the sentences and provide explanations for their classifications
  - ChatGPT vs Bryson:
    - ChatGPT: We use both GPT-3.5 and GPT-4 models (zero-shot)
      ⏩ Prompt
    - Bryson: 24-year-old Federal Reserve research assistant, holds a BSc, very bright, somewhat annoying
  - Focus on a few selected sentences (one from each classification group)

Summary of Reasoning Exercise



1. GPT models generally present a logic that successfully justifies classification choice
2. The explanations are very similar to those provided by Bryson, especially for GPT-4, with GPT-4 offering an improvement over GPT-3.5 with more cases of agreement with Bryson
3. GPT models not only outperform existing NLP methods, but offer a reasoning capability that existing methods cannot provide

How to Scale Qualitative Output?

LLMs generate qualitative insights – but how do we scale, structure, and analyze them systematically?

- Reasoning is often structured causally
- We can use causal frameworks to parse and interpret LLM outputs

Directed Acyclic Graphs (DAGs)

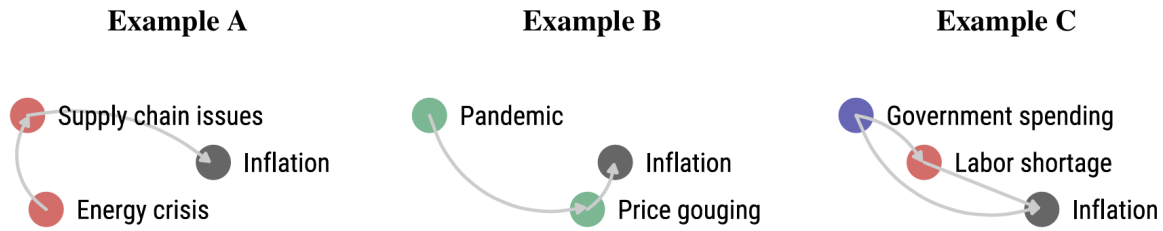
- Nodes = variables or concepts
- Edges = directional causal relationships
- *mixtape.scunning.com/03-directed_acyclical_graphs*

One can extract LLM causal claims from text and fit them into DAG form

How to Scale Qualitative Output?

"Narratives about the Macroeconomy" by Peter Andre, Ingar Haaland, Christopher Roth, Mirko Wiederholt, and Johannes Wohlfart

Figure 1: Example narratives, represented by DAGs



Notes: Three example narratives for why inflation increased, represented by their DAGs. Blue nodes are demand-side factors, red nodes are supply-side factors, and green nodes are miscellaneous factors. The arrows indicate the direction of causality.

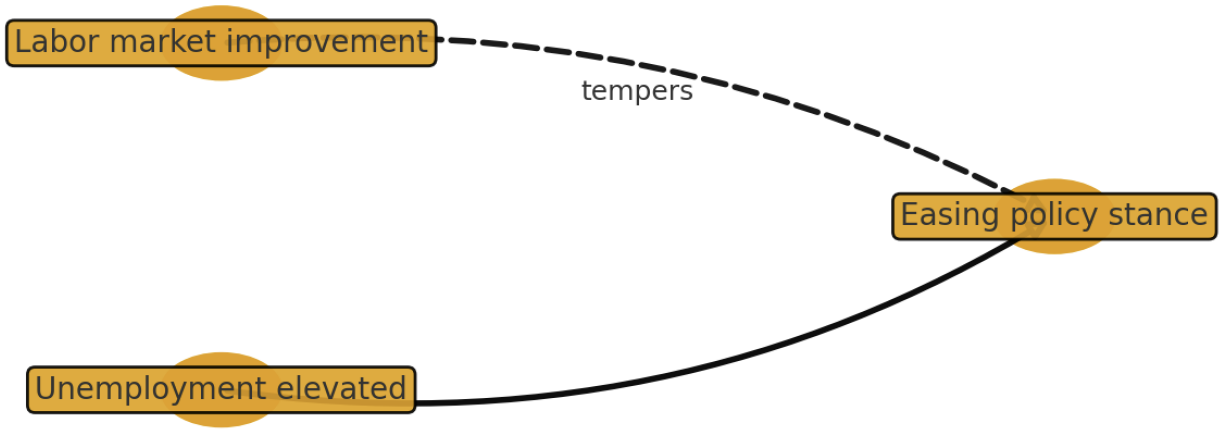
DAGs can bridge the gap between narrative and empirical analysis

Causal DAG Extraction Example

```
MODEL = "gpt-4o"
client = OpenAI()

SENTENCE = ("Labor market conditions have shown some improvement in recent months, "
            "on balance, but the unemployment rate remains elevated.")

SYSTEM = """You extract causal DAGs from monetary policy sentences.
Return ONLY JSON with fields:
{
 "nodes": [string, ...],
 "edges": [{"source": str, "target": str, "relation": "supports"|"tempers"}]
}"""

USER = f"""Sentence: {SENTENCE}
Rules:
- Nodes are concise economic signals or outcomes (e.g., "Unemployment elevated", "Easing policy stance").
- Add an outcome node for stance if implied (e.g., "Easing policy stance" or "Tightening policy stance").
- ‘supports’ means increases likelihood of target; ‘tempers’ means weakens/offsets it.
Keep <= 5 nodes, <= 6 edges.
"""

resp = client.responses.create(
    model=MODEL,
    input=[
        {"role": "system", "content": SYSTEM},
        {"role": "user", "content": USER},
    ],
    temperature=0,
    response_format={"type": "json_object"},
)
```

Sample DAG Diagram

"Labor market conditions have shown some improvement in recent months on balance, but the unemployment rate remains elevated."



The Narrative Approach

"It is hard to imagine that we could train a computer to read Federal Reserve transcripts the way we do. [...] We thoroughly expect to be made largely redundant by computers eventually, but perhaps not for a few years to come."

Christina D. Romer and David H. Romer (2023)

The Narrative Approach in Monetary Economics

- Narrative approach (Friedman 1963): Use qualitative sources to identify policy changes that were not motivated by factors affecting output
  - Requires identifying policy makers' motivation behind policy decisions
- Romer & Romer (1989): Implemented by manually reading through FOMC meeting material to identify monetary policy shocks
  - Reading through huge amounts of text $\sim$ 50-100 pages per meeting, 8-12 meetings per year, since 1946
- R&R (2023) provide an update to and revision of the 1989 paper

Policy Shock Definition

- A monetary policy shock refers to movement in monetary policy that is unrelated to current or prospective real economic activity
- A shock occurs when policy makers change money growth and interest rates due to concerns about prevailing inflation levels, even when the economy is stable
- Policy makers, in these instances, are willing to accept potential negative consequences for aggregate output

Digital Romer(s)

- Goal: reproduce R&R (1989, 2023) by letting GPT-4 read and interpret meeting materials
- Method:
  - Use the detailed transcripts from 1946–2017 and shorter minutes from 2017–2023, where transcripts are not available
  - Construct a prompt based on R&R's instructions and query the above documents
  - Use the GPT-4 model (with one example from R&R)

R&R Prompt

Narrative Approach Prompt

As a monetary policy expert, your task is to determine whether a given text contains a monetary policy shock. A monetary policy shock refers to movements in monetary policy that are unrelated to current or prospective real economic activity. These shocks occur when policymakers change money growth and interest rates due to concerns about prevailing inflation levels, even when the economy is stable. Policymakers, in these instances, are willing to accept potential negative consequences for aggregate output and unemployment.

Analyzing the provided text, determine whether it meets the criteria for a monetary policy shock based on the following factors:

- The policymakers believed the economy was at potential output.
- Policymakers changed money growth and interest rates due to high inflation.
- Policymakers understood and accepted the potential adverse consequences for output and unemployment.

Consider the example given as a reference:

Example: December 1988 "This episode counts as a contractionary monetary policy shock because, at a stable level of growth and unemployment, policymakers decided that the current level of inflation was unacceptable and took actions to reduce it. And they clearly understood and accepted that there could be substantial adverse consequences for output and unemployment."

Assess whether the provided text indicates a similar scenario. If it does, explain why it meets the criteria for a monetary policy shock. If it does not, provide a brief explanation of how it fails to meet the criteria.

Results: Can ChatGPT Identify R&R Policy Shocks?

- GPT-4 identifies most of the shocks discovered by R&R
- For the meetings where GPT-4 does not identify a shock but R&R do, we ask the model to explain why it didn't identify a shock
  - Absence of critical components in the definition of shocks; in most cases, missing evidence for the belief that the economy was operating at full potential

R&R (2023)	R&R (1989)	GPT-4
Oct. 1947	Oct. 1947	
Aug. 1955		Aug. 1955*
	Sept. 1955	Nov. 1955
Sept. 1958		
Dec. 1968	Dec. 1968	Dec. 1968
Apr. 1974	Apr. 1974	Apr. 1974*
Aug. 1978	Aug. 1978	
Oct. 1979	Oct. 1979	Oct. 1979
May 1981		May 1981*
Dec. 1988	Dec. 1988	Dec. 1988
June 2022		June 2022

Example: June 2022

Summary

- Despite impressive performance, LLMs are not infallible and can't fully replace human evaluators
- Any downstream use of LLM derived labels requires validation against gold standard data
- And, of course, qualified researchers are still needed to figure out what can be learned from the data, what type of data can be extracted, and in what shape

Financial Earning Calls

Evaluating Local Language Models:

An Application to Financial Earnings Calls *

Thomas R. Cook
Federal Reserve Bank of Kansas City

Anne Lundgaard Hansen
Federal Reserve Bank of Richmond

Sophia Kazinnik
Federal Reserve Bank of Richmond

Peter McAdam
Federal Reserve Bank of Kansas City

Abstract

This study evaluates the performance of local large language models (LLMs) in interpreting financial texts, in comparison to closed-source, cloud-based models. Our study is comprised of two main exercises. The first exercise benchmarks local LLM performance in analyzing financial and economic texts. Through this exercise, we introduce new benchmarking tasks for assessing LLM performance and explore the refinements needed to improve local LLM performance. Benchmarking results suggest that local LLMs are viable as a tool for general NLP analysis of financial and economic texts. In the second exercise, we leverage local LLMs to analyze the tone and substance of bank earnings calls in the post-pandemic era, including calls conducted during the banking stress of early 2023. Using local LLMs, we analyze remarks in bank earnings calls in terms of topics discussed, overall sentiment, temporal orientation, and vagueness. In response to the banking stress of early 2023, bank calls tended to converge to a similar set of topics and conveyed a distinctly less positive sentiment.

Keywords: Large Language Models (LLMs), Banking, Natural Language Processing (NLP)

Motivation

- Can local, open LLMs match closed models on domain-specific financial NLP tasks?
- Considerations: **reproducibility**, **data privacy**, **cost**, and **customization**
- Who cares? Well..

Motivation

- Can local, open LLMs match closed models on domain-specific financial NLP tasks?
- Considerations: **reproducibility**, **data privacy**, **cost**, and **customization**
- Who cares? Well.. CBs, financial firms, anyone working with PII or sensitive data..

Research Questions

1. How well do local LLMs perform on nuanced financial texts?
2. Which prompting techniques (e.g., CoT vs few-shot) boost accuracy?

Models Evaluated

- **Vicuna** (7B, 13B): instruction-tuned LLaMA
- **Wizard-Vicuna Uncensored** (30B)
- **Guanaco** (33B, 65B)
- **Fin-LLaMA** (33B)

Hardware-friendly setup via **4-bit quantization** (GPTQ) for most models.

Note: What is Quantization?

Quantization is the process of reducing the precision of the numbers used to represent model parameters

- Goal: Make models smaller, faster, and more hardware-friendly without significantly reducing performance.

4-Bit Quantization? (GPTQ)

- Reduces each weight to just 4 bits, allowing 8x compression compared to 32-bit float.
- Useful for running LLMs on consumer-grade GPUs or in constrained environments.

Another Note: GPU Poor?

How to Access Open Source Models?

There are several ways to access and run open source models.

Hosted model APIs

- `together.ai`
- `openrouter.ai`
- `deepinfra.com`

Rent GPU compute

- `runpod.io`

Research / inference orchestration

- `expectedparrot.com`

How to Access Open Source Models?

```python
from together import Together

client = Together() # auth defaults to os.environ.get("TOGETHER_API_KEY")

response = client.chat.completions.create(
    model="openai/gpt-oss-20b",
    messages=[
      {
        "role": "user",
        "content": "What are some fun things to do in New York?"
      }
    ]
)
print(response.choices[0].message.content)
```

Step 1: Performance Benchmarking

Benchmark 1: FOMC Stance Classification

- Task: label FOMC sentences as *hawkish* $\leftrightarrow$ *dovish* (3- and 5-class).
- Test different prompting approaches: **CoT**, **few-shot**.

Initial Prompt

Algorithm 1 Initial Prompt

SYSTEM: You are an economist. Read the following passage by a member of the FOMC {PASSAGE}
USER: Classify the passage into one of the 5 categories (hawkish, mostly hawkish, neutral, mostly dovish, dovish):
ASSISTANT: Rating: {CONSTRAINED LLM RESPONSE 1}
ASSISTANT: Explanation: {LLM RESPONSE 2}

CoT Prompt

Algorithm 2 COT Prompt

SYSTEM: You are an economist. A hawkish stance on monetary policy it favors higher interest rates to keep inflation in check. A dovish stance on monetary policy supports low interest rates to stimulate investment and spending and to maintain low unemployment.
USER: Read this passage from a speech by an FOMC member:
{PASSAGE}
Discuss the monetary policy stance that seems to be endorsed in the passage. Does it seem to be hawkish or dovish?
ASSISTANT: {LLM RESPONSE 1}
USER: Based on your response, rate it as hawkish, slightly hawkish, neutral, slightly dovish, or dovish
ASSISTANT: Rating: {CONSTRAINED LLM RESPONSE 2}

Few-shot Prompt

Algorithm 3 Few-Shot Prompt

USER: Rate this passage from a financial newspaper and score it's sentiment as positive, negative or neutral.
{EXAMPLE PASSAGE 1}
ASSISTANT: {EXAMPLE RATING 1}
USER: Rate this passage from a financial newspaper and score it's sentiment as positive, negative or neutral.
{EXAMPLE PASSAGE 2}
ASSISTANT: {EXAMPLE RATING 2}
USER: Rate this passage from a financial newspaper and score it's sentiment as positive, negative or neutral.
{PASSAGE}
ASSISTANT: {CONSTRAINED LLM RESPONSE 1}

Benchmark 1: FOMC Stance Classification

- Task: label FOMC sentences as *hawkish* $\leftrightarrow$ *dovish* (3- and 5-class); use human labeled data from previous paper

Table 2: Model performance with 3- and 5-categories using initial prompt, chain of thought (COT) prompting, and few-shot prompting

	3-category accuracy			5-category accuracy		
Model	Initial	COT	Few-shot	Initial	COT	Few-shot
Wizard-Vicuna	0.47	0.69	0.83	0.30	0.39	0.53
Guanaco (33B)	0.16	0.70	0.60	0.14	0.41	0.40
Guanaco (65B)	0.62	0.84	0.80	0.30	0.38	0.61
Fin-LLaMA	0.16	0.80	0.40	0.02	0.46	0.29
Vicuna (7B)	0.04	0.80	0.78	0.04	0.34	0.46
Vicuna (13B)	0.80	0.77	0.81	0.48	0.32	0.40

Notes: Scores reported under the Few-Shot column reflect the best performance for the model across a battery of few-shot prompts. The complete list of accuracy scores for the few-shot prompts used in this exercise can be found in appendix B.1.

Benchmark 2: Financial Phrase Bank (Malo et al. 2014)

huggingface.co/datasets/takala/financial_phrasebank

Benchmark 2: Financial Phrase Bank (Malo et al. 2014)

huggingface.co/datasets/takala/financial_phrasebank

- 4,840 English-language sentences sourced from financial news and press releases.
- Sentiment annotations: **positive**, **neutral**, or **negative**; multiple annotators per sentence.
- We focus on the **75% agreement** subset (i.e., $\geq$75% inter-annotator consensus), for cleaner training data.

Benchmark 2: Financial Phrase Bank (Malo et al. 2014)

huggingface.co/datasets/takala/financial_phrasebank

- 4,840 English-language sentences sourced from financial news and press releases.
- Sentiment annotations: **positive**, **neutral**, or **negative**; multiple annotators per sentence.
- We focus on the **75% agreement** subset (i.e., $\geq$75% inter-annotator consensus), for cleaner training data.
- We also augment this benchmark with two additional annotation dimensions on 1,000 sentences:
  - **Vagueness:** Label each sentence as clear or vague to capture linguistic ambiguity in financial language.
  - **Temporality:** Classify whether the sentence refers to the past, present, or future, enabling temporal grounding.

Extended Annotations: Vagueness

Definition 1: Vagueness

A sentence is either clear or vague.

With this label we aim to identify language that intentionally convolutes the message or avoids addressing the subject directly.

- A vague sentence can be subject to misinterpretation or multiple interpretations.
- A clear sentence is not subject to either.
- A vague sentence can convey contradictory messages.
- A clear sentence does not.

Extended Annotations: Temporality

Definition 2: Temporality

A sentence is focused on the past, the present, or the future.

- Backward-looking text discusses history, interprets past events, or reflects on prior outcomes.
- Present-focused text interprets or comments on ongoing events.
- Forward-looking text refers to the future, including plans, expectations, or strategies.

Phrase Bank Evaluation

We evaluate the ability of the considered LLMs to classify the sentiment of Phrase Bank sentences along the three dimensions:

(i) positive/neutral/negative,

(ii) clarity/vagueness, and

(iii) past/present/future

Phrase Bank: Sentiment Results (Few-shot)

Table 4: Model performance with few-shot prompting for classifying positive–neutral–negative sentiment of Phrase Bank sentences

	Wizard-Vicuna	Guanaco (33B)	Guanaco (65B)	Fin-LLaMA	Vicuna (7B)	Vicuna (13B)
Accuracy	0.766	0.784	0.694	0.784	0.798	**0.826**
Precision:						
Positive	0.587	**0.891**	0.468	0.833	0.815	0.717
Neutral	0.839	0.756	**0.962**	0.763	0.779	0.854
Negative	**1.000**	0.968	0.904	**1.000**	**1.000**	0.932
Recall:						
Positive	0.766	0.383	**0.969**	0.508	0.516	0.711
Neutral	0.784	**0.981**	0.552	0.959	0.953	0.881
Negative	0.660	0.566	**0.887**	0.396	0.547	0.774
Specificity:						
Positive	0.766	0.383	**0.969**	0.508	0.516	0.711
Neutral	0.784	**0.981**	0.552	0.959	0.953	0.881
Negative	0.660	0.566	**0.887**	0.396	0.547	0.774
F_1 score:						
Positive	0.664	0.536	0.631	0.631	0.632	**0.714**
Neutral	0.810	0.854	0.701	0.850	0.858	**0.867**
Negative	0.795	0.714	**0.895**	0.568	0.707	0.845
Balanced accuracy:						
Positive	0.785	0.683	0.791	0.735	0.736	**0.805**
Neutral	0.759	0.710	0.756	0.717	0.739	**0.807**
Negative	0.830	0.782	**0.935**	0.698	0.774	0.883

Notes: The table reports classification performance for the subset of Phrase Bank sentences for which at least 75% annotators agree (a total of 3,448 sentences). The considered LLMs are implemented with few-shot prompting. **Bold-faced** values indicate the winning model per performance metric.

Phrase Bank: Vagueness Detection

Table 5: Model performance for classifying clarity of Phrase Bank sentences

	Wizard-Vicuna	Guanaco (33B)	Guanaco (65B)	Fin-LLaMA	Vicuna (7B)	Vicuna (13B)
Accuracy	0.861	0.725	0.696	0.839	**0.963**	0.846
Precision:						
Clear	0.990	0.992	0.994	0.971	0.974	**0.995**
Vague	0.228	0.133	0.124	0.139	**0.640**	0.221
Recall:						
Clear	0.863	0.718	0.686	0.857	**0.987**	0.843
Vague	0.824	0.882	**0.912**	0.471	0.471	**0.912**
F_1 score:						
Clear	0.922	0.833	0.812	0.910	**0.981**	0.913
Vague	0.357	0.231	0.219	0.215	**0.542**	0.356

Notes: The table reports classification performance for the subset of the 1000 sentences for which at least 75% of the annotators agree (a total of 760 sentences). **Bold-faced** values indicate the winning model per performance metric.

- Highly **imbalanced** (few "vague" cases).
- **Vicuna-7B**: best trade-off; detects around 50% of vague sentences.
- Agreement-sensitive: when annotators **all agree**, models capture vagueness

Phrase Bank: Temporality Classification

Table 6: Model performance for classifying temporality of Financial Phrase Bank sentences

	Wizard-Vicuna	Guanaco (33B)	Guanaco (65B)	Fin-LLaMA	Vicuna (7B)	Vicuna (13B)
Accuracy	0.746	0.706	**0.767**	0.745	0.605	0.671
Precision:						
Positive	0.767	0.632	0.851	0.824	0.853	**0.905**
Neutral	0.674	0.717	0.732	**0.825**	0.527	0.709
Negative	**0.828**	0.750	0.771	0.673	0.698	0.618
Recall:						
Positive	0.573	**0.596**	0.484	0.507	0.136	0.268
Neutral	**0.939**	0.883	**0.939**	0.781	0.951	0.709
Negative	0.711	0.639	0.826	0.882	0.652	**0.921**
Specificity:						
Positive	0.924	0.848	0.964	0.953	**0.989**	0.987
Neutral	0.752	0.789	0.807	**0.902**	0.519	0.824
Negative	**0.887**	0.841	0.817	0.697	0.754	0.571
F_1 score:						
Positive	**0.656**	0.614	0.617	0.628	0.235	0.413
Neutral	0.785	0.791	**0.823**	0.802	0.678	0.709
Negative	0.765	0.690	**0.797**	0.763	0.675	0.739
Balanced accuracy:						
Positive	**0.748**	0.722	0.724	0.730	0.562	0.627
Neutral	0.845	0.836	**0.873**	0.842	0.735	0.766
Negative	0.799	0.740	**0.822**	0.789	0.703	0.746

Notes: The table reports classification performance for the subset of the 1000 sentences for which at least 75% of the annotators agree (a total of 769 sentences). **Bold-faced** values indicate the winning model per performance metric.

Corpus: Bank Earnings Calls (2021–2023Q2)

Each earnings call is about 12,000 words long (around 50 pages)

We separate each transcript into smaller parts corresponding to a passage of unbroken text spoken by an individual speaker, i.e., **utterance level**

- Over 100 banks per quarter (LISCC + large/regional/community).
- Goal is to assign each passage with: sentiment, clarity, temporality, and *topics*.

Topic Modeling Pipeline (LLM + Embeddings)

Algorithm 4 Topic Model Prompt

USER: Read the following passage and describe it in terms of a broader set of categories:
{PASSAGE}
ASSISTANT:
Topics:
1. {LLM RESPONSE 1}
2. {LLM RESPONSE 2}
3. {LLM RESPONSE 3}
4. {LLM RESPONSE 4}
5. {LLM RESPONSE 5}

1. LLM outputs up to 5 **free-text categories** per passage, of restricted length

The Problem: Inconsistent Topic Labels

We do not restrict the possible responses of the LLM, the categories identified tend to be similar but still vary between passages:

- LLM generates 5 topic labels per passage:
  - "Real Estate Market"
  - "Housing Sector"
  - "Residential Properties"
- All refer to the same concept – but use different wording
- Without clustering, they're treated as separate topics

Goal: Group semantically similar topics for analysis

The Solution: Sentence Embeddings + K-Means

1. **Sentence Embeddings**
   - Convert topic labels into high-dimensional vectors
   - Use pre-trained model: `all-mpnet-base-v2`
   - Similar meanings $\rightarrow$ nearby vectors
2. **K-Means Clustering**
   - Group vectors into K clusters (e.g., $K = 150$)
   - Each cluster = a generalized topic

Example: "Housing Market" and "Real Estate" $\rightarrow$ Cluster 27: `Housing`

LLM Topic Modeling: Full Pipeline

1. LLM generates 5 short topic labels per passage
2. Convert each label into a sentence embedding
3. Cluster embeddings using K-means
4. Assign clustered topic IDs to each passage

Benefits:

- Aggregates semantically similar labels
- Enables robust topic frequency and similarity analysis
- Supports tracking trends over time (e.g., banking stress response)

Illustrative Topics (Top Frequencies)

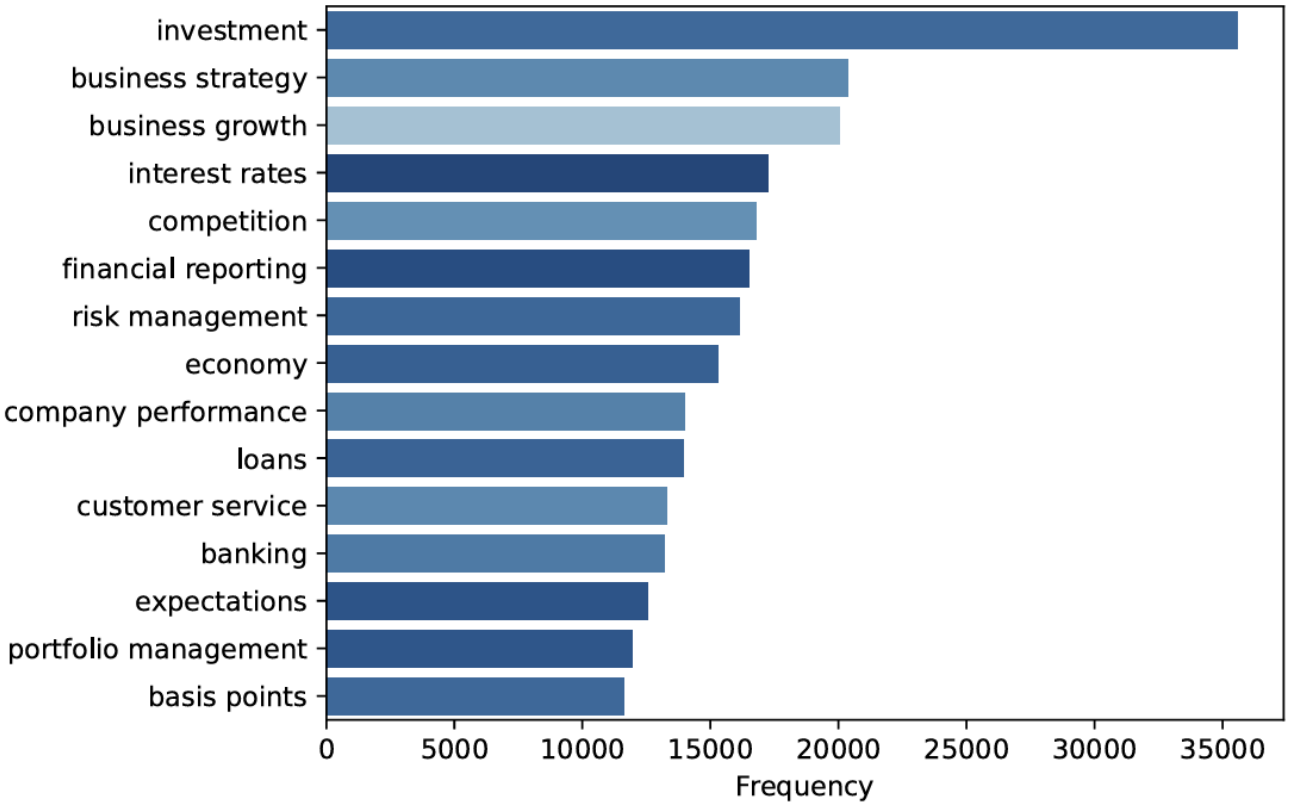
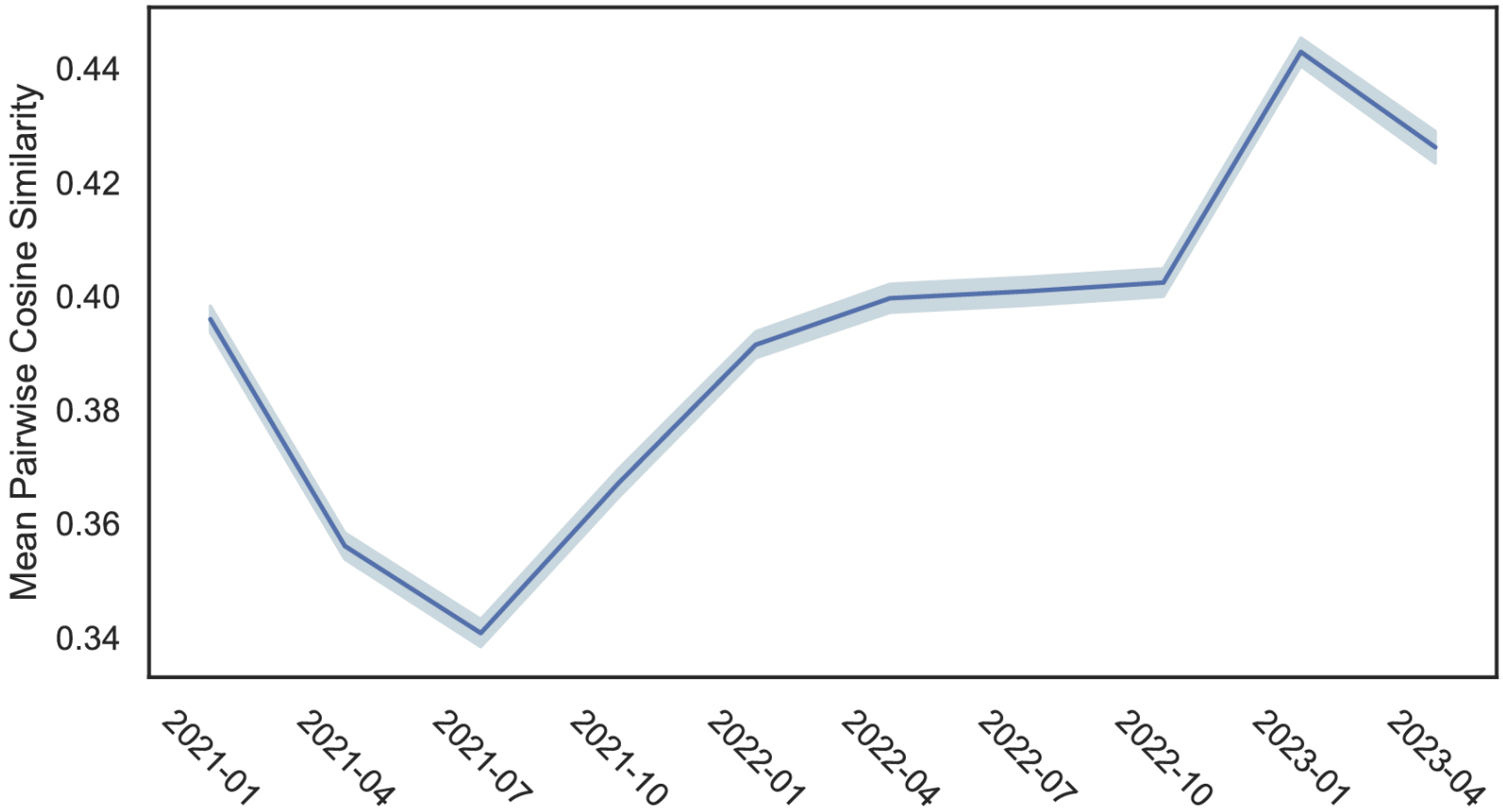


FIGURE 1. Most Prevalent Topics Across All Earnings Calls

NOTES: *The figure shows the estimated topics that appear most frequently in the full sample of earnings call transcripts. The color indicates the average Sentiment with which each topic is discussed, where lighter color refers to more positive sentiment, and darker color is associated with more negative sentiment.*

Crisis Signal: Topic Convergence (2023Q1)

Figure 3: Similarity in topics discussed



- Peak similarity in 2023Q1 (SVB stress) $\Rightarrow$ banks *reassure* on common risks.

Deposits Topic Dominates During Stress

- Relative frequency of **deposit** topic jumps **200%+** in 2023Q1.
- Similar increases for **risk management**; **interest rates** rises from late 2021 onward.

Tone Over Time: Sentiment, Clarity, Temporality

- **Sentiment:**
  - Peaks in 2021 during low-stress period
  - Falls sharply in 2023Q1, following SVB and FRCB collapses
  - Suggests shift from promotional to reassurance tone
- **Temporality:**
  - Gradual shift from forward-looking (2021) to present-focused (2023)
  - Reflects change from optimism to crisis response
- **Clarity:**
  - Largely stable across time
  - Indicates banks did not become more vague, even under stress

Interpretation: Patterns consistent with signaling model: banks promote in calm periods, reassure in crises.

Topic-Level Sentiment: Deposits & Interest Rates

- **Sentiment trends by topic:**
  - **Deposits:** Highly positive during low-stress 2021; sharp decline by 2023Q1
  - **Interest Rates:** Similar pattern – optimism fades as Fed hikes begin
- **Interpretation:**
  - Banks shift tone as deposit fragility and duration risk grow salient
  - Reinforces the *pooling-on-reassurance* behavior from the signaling game

NLP Takeaways

- **Local LLMs are viable:**
  - Strong performance with prompt refinement
  - Suitable for private and domain-specific analysis
- **Prompting matters:**
  - **Few-shot** outperforms COT for financial sentiment (Phrase Bank)
  - **COT** improves classification of policy tone (hawkish/dovish)
- **Data challenges:**
  - Vagueness classification suffers from label imbalance
  - Requires calibrated thresholds or clearer annotation guidelines
- **Topic modeling pipeline:**
  - Simple LLM $\rightarrow$ embedding $\rightarrow$ clustering approach
  - Scales well to full-length earnings call transcripts

Model Explainability

Model-Agnostic Explainability Methods

"..*the degree to which a human can understand the cause of a decision*"

- 1. LIME (Local Interpretable Model-agnostic Explanations):
  - Perturbs inputs and observes changes in output.
  - Fits a simple interpretable model (e.g., linear) around a prediction.
  - Highlights which input features influenced the decision most.
- 2. SHAP (SHapley Additive exPlanations)
  - Based on cooperative game theory (Shapley values).
  - Measures the contribution of each input feature to the output (masking).
  - Can be applied to token-level or feature-level explanations in NLP.
- 3. Partial Dependence Plots (PDP)
- 4. Many other algorithms..

These methods work with any model, including black-box LLMs

What is a Partial Dependence Plot (PDP)?

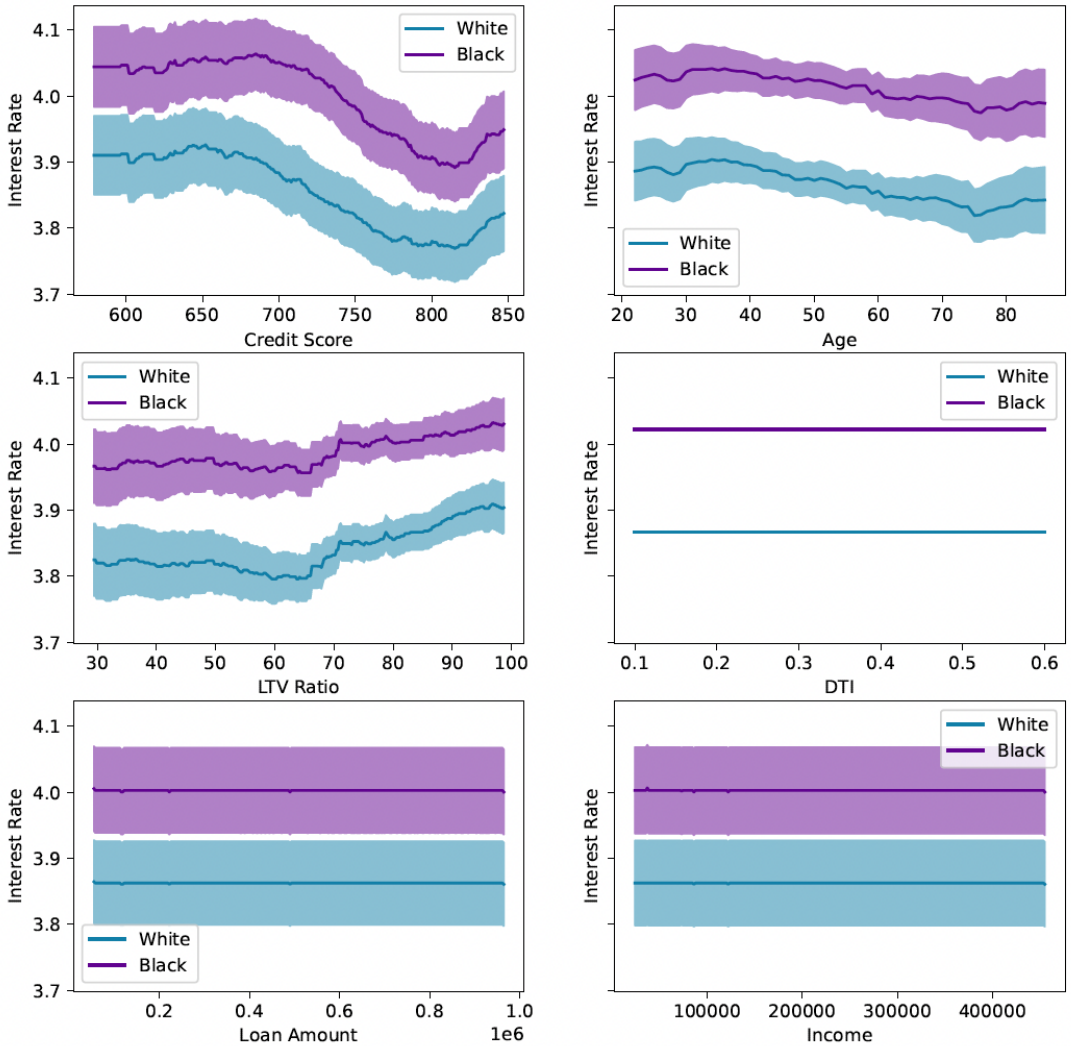
Goal: Understand how the model's prediction varies with a feature, on average over the other observed features.

- PDP answers:

  "If I only change this one input, how does the model's prediction change on average?"
- Inspired by the economics idea of **ceteris paribus**.
- For each value of a variable (e.g., credit score):
  - Keep all other features as they are (from actual data).
  - Compute model predictions.
  - Average across all observations.
- Repeat across a grid of values to build the curve.

Partial Dependence Plots (PDPs)

Figure 3: PDP-GP for each applicant feature for the Mistral v0.3 Model



PDP Example: Credit Score → Predicted Interest Rate

- Steeper slope ⇒ model is sensitive to credit score.
- Vertical gap ⇒ systematic difference between groups.

An Applied Exercise

Goal: Sentiment Labeling with LLMs

- Use open-source LLMs via Together.ai to label financial text.
- Labels: `positive`, `neutral`, `negative`, `uncertain`

Goal: Sentiment Labeling with LLMs

- Dataset: Finance PhraseBank (via Hugging Face)
- Compare performance across multiple models.

Step 1: Load the Dataset from Hugging Face

```
from datasets import load_dataset

dataset = load_dataset("financial_phrasebank", "
    sentences_allagree")
df = dataset["train"].to_pandas()
df.columns = ['text', 'label']
df['label'] = df['label'].str.lower()
df.sample(3)
```

Step 2: Build the Prompt

```
def make_prompt(text):
    return f"""
Classify the sentiment of the following financial sentence.

Label it as one of: "positive", "negative", "neutral", or "
    uncertain".

Return a JSON object with one key: "label".

Sentence:
\"\"\"{text}\"\"\"
"""
```

Step 3: Set Up Together.ai API

```
!pip install together

import together
together.api_key = "your-api-key"
```

Models we'll use:

- `mistralai/Mistral-7B-Instruct-v0.1`
- `mistralai/Mixtral-8x7B-Instruct-v0.1`
- `meta-llama/Llama-2-7b-chat-hf`

Step 4: Call LLMs via API

```python
def call_model(model_name, prompt):
    response = together.Complete.create(
        model=model_name,
        prompt=prompt,
        max_tokens=100,
        temperature=0
    )
    return response['output']['choices'][0]['text']
```

Step 5: Apply Models and Parse JSON

```
import json

models = [
    "mistralai/Mistral-7B-Instruct-v0.1",
    "mistralai/Mixtral-8x7B-Instruct-v0.1",
    "meta-llama/Llama-2-7b-chat-hf"
]

def label_with_models(text):
    prompt = make_prompt(text)
    results = {}
    for model in models:
        try:
            raw = call_model(model, prompt)
            parsed = json.loads(raw.strip())
            results[model] = parsed.get("label", "error")
        except:
            results[model] = "error"
    return results
```

Step 6: Run and Analyze

```python
df_sample = df.sample(20).copy()
df_sample["model_outputs"] = df_sample["text"].apply(
    label_with_models)

# Flatten results
model_df = df_sample["model_outputs"].apply(pd.Series)
final_df = pd.concat([df_sample["text"], df_sample["label"],
     model_df], axis=1)

# Accuracy
for model in models:
    acc = (final_df[model] == final_df["label"]).mean()
    print(f"{model}: {acc:.2%}")
```

Questions?

Day 2 wrap-up