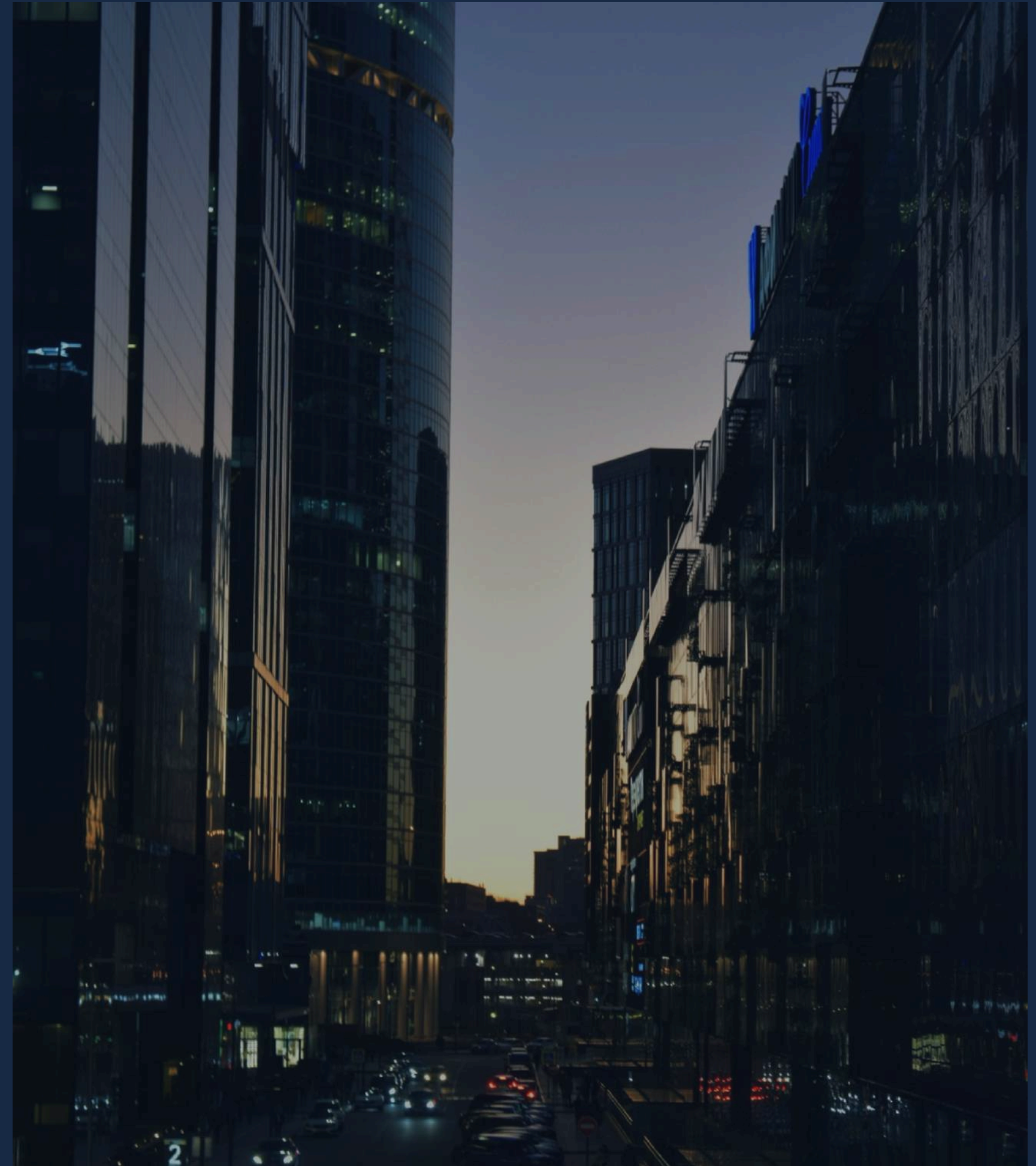




EARNINGS CALL ANALYZER

INTRODUCTION

Developed an NLP Tool for extracting financial metrics and generating trading signals





PROBLEM

- Earnings calls include valuable signals for traders such as guidance, sentiment and growth drivers.
- Data is unstructured, in language form, time-consuming, and inconvenient to analyze and access.
- Investors and analysts need fast, consistent, and actionable insights.

SOLUTION

Built a CLI-based Python tool that:

- Parses earnings call transcripts
- Extracts key financial metrics
- Applies NLP sentiment analysis
- Generates BUY / SELL / HOLD recommendations
- Outputs analyst-ready CSV files

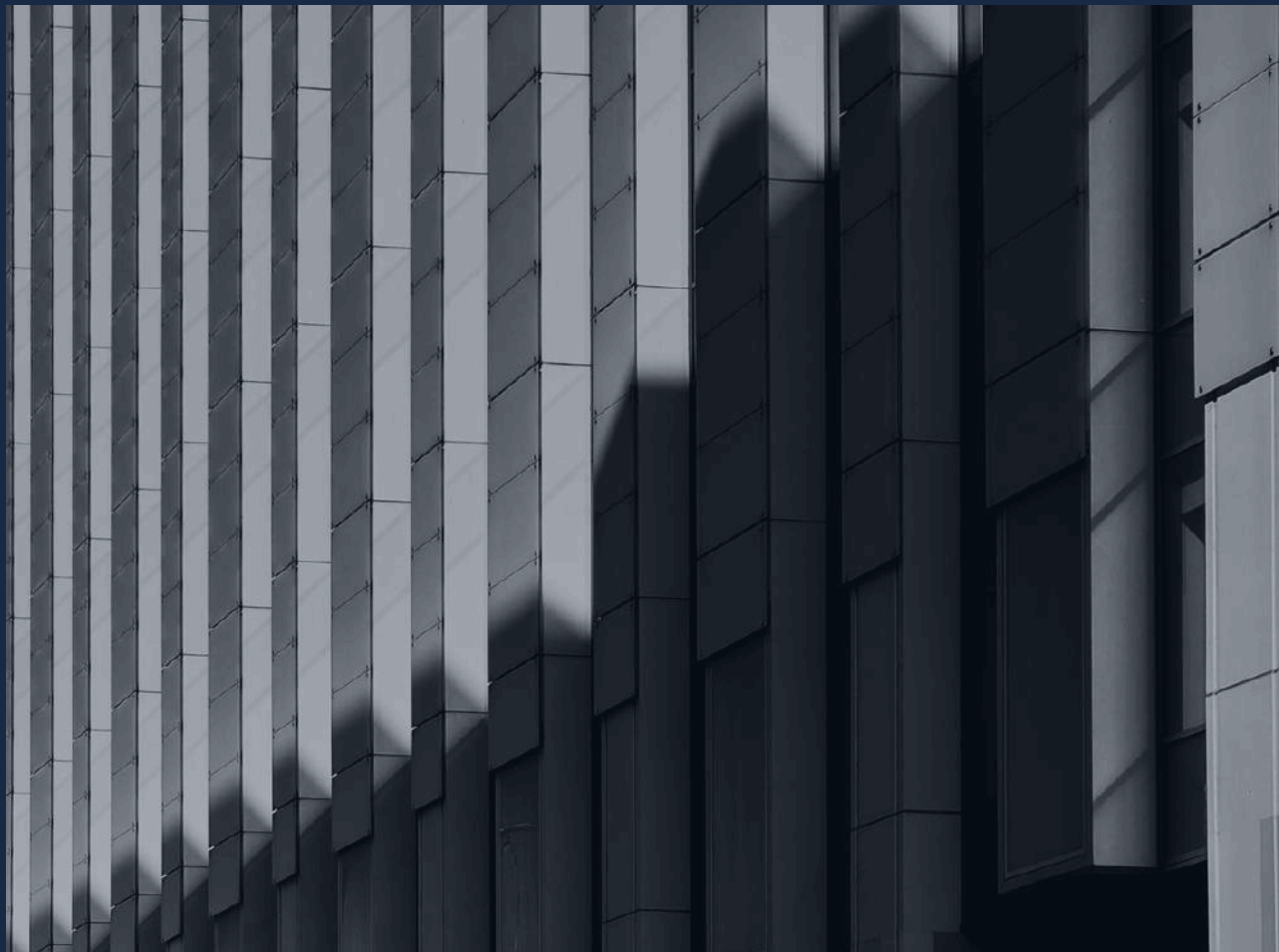
Designed as an end-to-end NLP + data engineering pipeline.





DATA

- Earnings call transcripts sourced from a public dataset on Kaggle
- Multiple companies and quarters
- Raw text used as model input



EXAMPLE OUTPUT

Input

- AMD Q4 2017 transcript

Output

- Revenue: \$1.48B (+34% YoY)
- Guidance: Maintained
- Sentiment: Positive

Recommendation: BUY (High confidence)

Strategy: Buy calls

Exports results to CSV for analysis.

METHODOLOGY

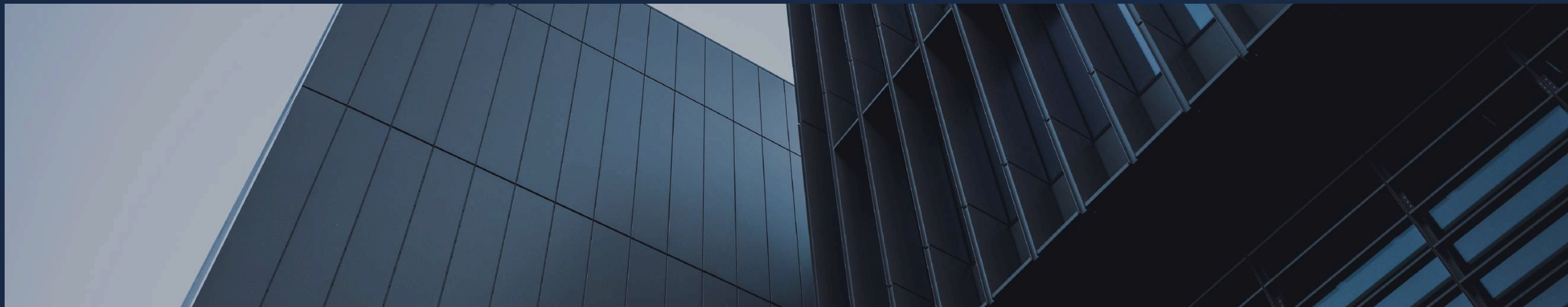
1. Parse transcript text
2. Financial sentiment via FinBERT
3. Regex + NLP pattern extraction for metrics
4. Rule-based scoring engine
5. Final recommendation + confidence level

Primary decision signals:

- Forward guidance
- Revenue growth
- Margin trends
- Management tone



RECOMMENDATION LOGIC



Strong Revenue Growth = Positive Signal

Positive Management Sentiment = Positive Signal

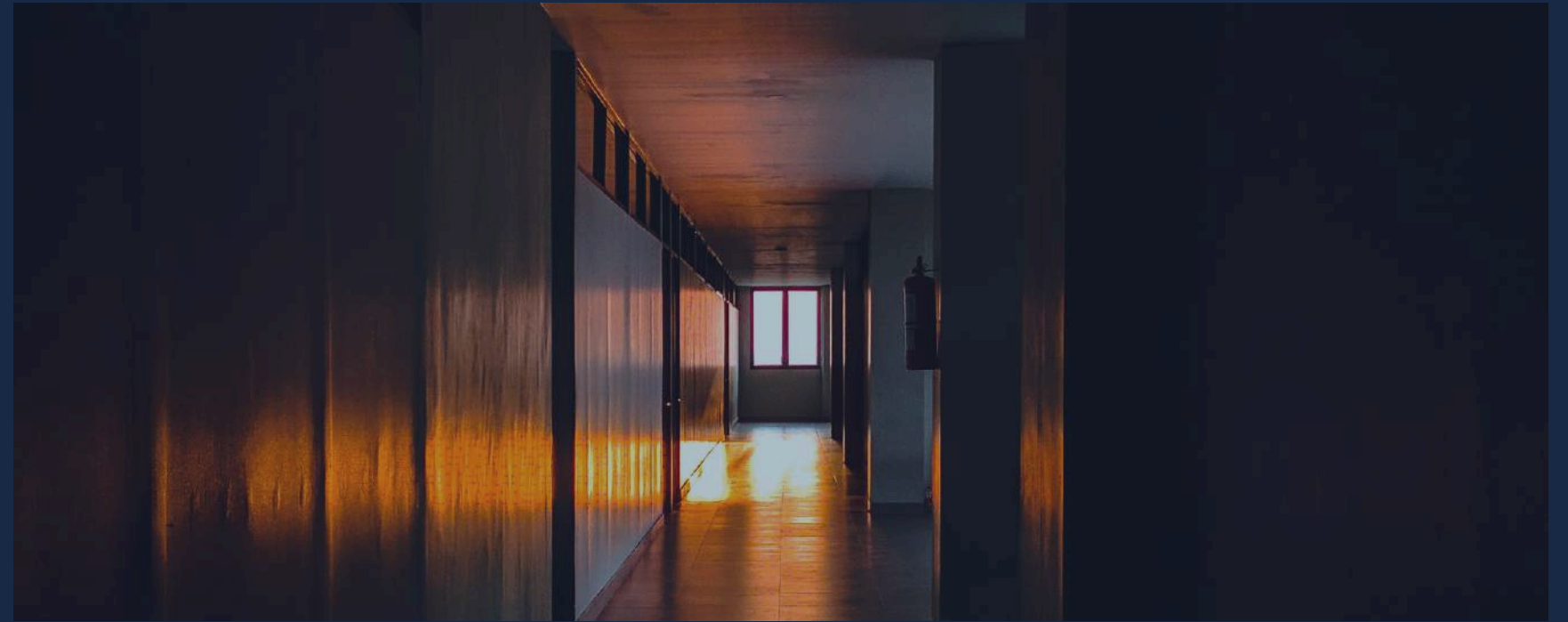
Guidance direction weighted heavily

Combined score
determines:

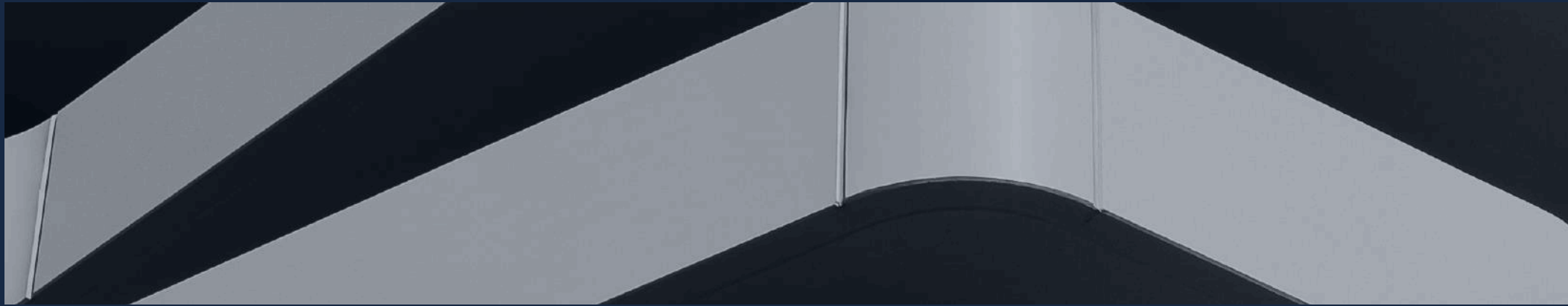
- BUY
- SELL
- HOLD

TECH STACK

- Python
- Hugging Face Transformers (FinBERT)
- PyTorch
- pandas/numpy
- CLI Interface + CSV export



WHAT THIS DEMONSTRATES



- NLP on real financial documents
- ML model integration
- Unstructured to structured data workflows
- Feature engineering from text
- Decision systems backed by domain knowledge



THANK
YOU