MGMT 638

Session 8

Kerry Back

Fall 2025

Agenda

- More discussion of AI
- Backtesting a PE prediction trading model
  - Variables suggested by Gordon and DuPont models
  - Size classifications
  - Backtesting process
  - LightGBM model
  - Results
  - Exploring fitted LightGBM trees

JPMorgan: Chatbots to Agents

CNBC, September 30, 2025: In 2023, JPMorgan gave employees access to OpenAI's models through LLM Suite.

It was essentially a **corporate ChatGPT tool** used to draft emails and summarize documents.

About 250,000 JPMorgan employees have access to the platform today … Half of them use it roughly every day.

JPMorgan is now early in the next phase of its AI blueprint: It **has begun deploying agentic AI** to handle complex multistep tasks for employees, according to an internal road map provided by the bank.

From Makers to Checkers

Derek Waldron, JP Morgan Chief Analytics Officer:

> What we're working towards is that every employee will have their own personalized AI assistant; every process is powered by AI agents, and every client experience has an AI concierge.
>
> You'll still have people at the top who are managing and have relationships with clients, but many, many of the processes underneath are now being done by AI systems.
>
> Workers would shift from being creators of reports or software updates, or 'makers' ... to 'checkers' or managers of AI agents doing that work.

Mathematical exploration and discovery at scale

Recent paper by Terence Tao and co-authors (Nov 3, 2025):

> *Large language models . . . can discover explicit constructions that either match or improve upon the best-known bounds to long-standing mathematical problems, at large scales . . . We considered a list of 67 problems spanning mathematical analysis, combinatorics, geometry, and number theory. The system rediscovered the best known solutions in most of the cases and discovered improved solutions in several.*

Using Apps or .py Files with AI

- Where possible, we should permanently fix the code used for tasks rather than asking AI to regenerate it each time.
- This ensures consistent behavior and also reduces token usage.
- We can create apps or, easier, save as a .py file and use in Claude Code skill.
- Examine your .claude\skills\rice-data-query folder.

Gordon Growth Model Formulas

Gordon Growth Model (Dividend Discount Model):

$$P_0 = \frac{D_1}{r - g} \quad \Leftrightarrow \text{PE} = \frac{(1+g) \times \text{Payout Ratio}}{r - g}$$

where P_0 is the current stock price, D_1 is the expected dividend next period, r is the required return, and g is the constant growth rate.

Sustainable Growth Rate:

$$g = \text{ROE} \times \text{Plowback Ratio} = \text{ROE} \times (1 - \text{Payout Ratio})$$

DuPont Formula for ROE:

$$\text{ROE} = \frac{\text{Net Income}}{\text{Sales}} \times \frac{\text{Sales}}{\text{Assets}} \times \frac{\text{Assets}}{\text{Equity}}$$
$$= \text{Profit Margin} \times \text{Asset Turnover} \times \text{Equity Multiplier}$$

Firm Size Classification by Market Capitalization

Industry-Standard Size Categories:

- **Mega-Cap:** Market cap $\geq$ \$200 billion
- **Large-Cap:** Market cap \$10 billion – \$200 billion
- **Mid-Cap:** Market cap \$2 billion – \$10 billion
- **Small-Cap:** Market cap \$300 million – \$2 billion
- **Micro-Cap:** Market cap \$50 million – \$300 million
- **Nano-Cap:** Market cap $<$ \$50 million

Distribution in November 2025 (3,261 firms):

- Mega-Cap: 1.47% Large-Cap: 19.93% Mid-Cap: 27.14%
- Small-Cap: 32.63% Micro-Cap: 15.49% Nano-Cap: 3.34%

Implementation: Applied these percentages each month to classify

Data and Scripts

Download Zip File

- Data from the Rice database: data3.parquet, data3_returns.parquet
- Output files created by scripts: lightgbm_pe_predictions.parquet, data3_evaluate.parquet, data3_portfolios.csv
- Scripts: fetch_monthly_returns_all.py, train_lightgbm_pe.py
- Jupyter notebook: pe_prediction_analysis.ipynb

data3.parquet

- Source: data2.parquet (filtered for positive PE ratios and with some other ratios)
- 338,352 rows, 29 columns
- Ticker, month, pe
- Categorical features: size, sector, industry
- Profitability ratios: roe, roa, grossmargin, netmargin, gp_to_assets
- Other ratios: assetturnover, equity_multiplier,
- Growth rates: many 5-year growth rates

Backtesting Process: Walk-Forward Validation

The Challenge: We want to predict PE ratios and trade on prediction errors, but we can only use past information

The Solution: Walk-Forward Monthly Backtesting

1. **Train:** Use data from month t to train model
2. **Predict:** Use trained model to predict PE ratios for month $t + 1$
3. **Form Portfolios:** At end of month t, rank stocks by prediction error (predicted PE - actual PE) / actual PE
4. **Calculate Returns:** Measure returns during month $t + 1$
5. **Continue:** Move to next month, retrain model with new data

Key Principle: Always use current features (known at time t) and a model trained on past data to predict future PE ratios and form portfolios

LightGBM: Light Gradient Boosting Machine

What is LightGBM?

- A gradient boosting framework that uses tree-based learning algorithms
- Builds an ensemble of decision trees sequentially, where each tree corrects errors from previous trees
- Optimized for speed and efficiency, especially with large datasets

Key Feature: Native Support for Categorical Variables

- LightGBM can handle categorical features directly without one-hot encoding (i.e., without dummy variables)
- Finds optimal splits by grouping categories intelligently
- For example, with a "sector" variable, it might group Technology, Healthcare, and Consumer sectors together if they have similar target values

Why Use LightGBM?

- Handles mixed data types: numeric ratios (ROE, margins) + categorical (sector, industry, size)
- Fast training and prediction, even with hundreds of thousands of observations

data3_returns.parquet

- Source: Rice Data Portal SEP table (Jan 2010 - present)
- 876,377 rows, 9,335 unique tickers
- Contains: ticker, month, close (end of prior month), return (current month)
- Created by: fetch_monthly_returns_all.py

train_lightgbm_pe.py

- Monthly walk-forward training and prediction
- Uses 500 trees, MAPE objective, learning rate 0.05
- Features: 3 categorical (sector, industry, size) + 23 numeric ratios
- Output: lightgbm_pe_predictions.parquet

lightgbm_pe_predictions.parquet

- 141,676 rows, 6 columns
- Columns: ticker, train_month, test_month, actual_pe, predicted_pe, percentage_error
- Contains all predictions from walk-forward validation (2015-09 to 2025-11)
- Used to create data3_evaluate.parquet by merging with returns data

Prediction and Portfolio Files

data3_evaluate.parquet

- Merged predictions with returns data
- Filtered for close $\geq$ \$5.00
- 141,676 rows with predictions
- Contains: ticker, month, close, return, pct_error

data3_portfolios.csv

- Decile portfolios based on pct_error
- 123 months (rows) $\times$ 10 deciles (columns)
- Mean returns: Decile 1: 0.99%, Decile 10: 1.12%

Jupyter Notebook: pe_prediction_analysis.ipynb

Interactive Analysis of November 2025 Predictions

What's included:

- Trains LightGBM on October 2025 data to predict November 2025 PE ratios
- Shows distribution of percentage errors with histogram and box plot
- Displays feature importance (top 20 features)
- Visualizes tree structure for two trees (Tree 0 and Tree 50)
- Actual vs Predicted scatter plot with R-squared
- Lists best and worst predictions by ticker

Purpose: Provides hands-on exploration of how LightGBM makes predictions, which features matter most, and how the ensemble of trees combines to form predictions