Term Project

Greg Roberts and Flavio Mota

MSDS 459 - Knowledge Engineering

June 8, 2025

Abstract

This paper documents the development and application of a Neo4j-based knowledge graph to support predictive modeling in the consumer discretionary sector. Moreover, this project discusses the importance of sentiment analysis in predicting stock market movement, especially using media measures. The knowledge base integrates Wikipedia company specific information, Yahoo Finance stock data and news articles, Fiserv's Small Business Index (FSBI), and macroeconomic indicators from FRED. During the final phase of the project, the pipeline was migrated to Google Colab using Neo4j AuraDB to enable cloud-based analytics. Sentiment analysis was performed on Yahoo Finance articles using the FinBERT transformer, and sentiment scores were integrated into an XGBoost model to predict daily stock returns over a 30-day sample period. The final model achieved a mean accuracy of 0.903 ± 0.017, F1 score of 0.907 ± 0.016, and ROC AUC of 0.969 ± 0.010. Feature importance analysis revealed that the primary predictive signals were derived from lagged stock returns, with sentiment features contributing but at a lower relative importance. These results demonstrate the value of combining knowledge graphs and sentiment analysis with historical market data for competitive intelligence and stock forecasting.

Introduction

The consumer discretionary sector is marked by rapid changes in consumer preferences and competitive dynamics, making it a challenging environment for investment analysis. Traditional relational databases often struggle to represent the complex, interconnected nature of companies, products, and macroeconomic indicators in this sector. This project addresses that gap by constructing a Neo4j-based knowledge graph that unifies textual and structured data sources into a single, flexible platform. While earlier work focused on backend data integration, the final phase emphasized operationalizing the knowledge base in a predictive model, leveraging features derived from the graph—including sentiment measures from Yahoo Finance articles—to enhance stock return forecasts. This paper details the entire project lifecycle, including

insights from migrating to Google Colab and using AuraDB for cloud-based graph storage.

Literature Review

Foundational work by Kejriwal, Knoblock, and Szekely (2021) describes semi-automated pipelines for relation extraction and graph construction, illustrating how knowledge graphs can bind heterogeneous data into a coherent structure. In financial applications, Yao et al. (2018) demonstrated the use of graph embeddings and traversal techniques for risk analysis, while Paulheim (2017) explored recommendation systems built on graph representations of corporate and market data. These studies highlight the benefits of graph queries for uncovering non-obvious correlations, such as linking consumer-spending anomalies to specific corporate events and informed our choice of Neo4j for its flexible schema and rich query language.

Methods

We began by scraping Wikipedia infoboxes for 10 consumer discretionary companies using BeautifulSoup. Parsed fields included company name, headquarters, industry, and product lines, which were normalized and stored in Neo4j using a custom DocumentParser class. Yahoo Finance data was ingested via the yfinance client, capturing daily closing prices and monthly returns. FSBI metrics were loaded from CSV exports and stored as FSBIIndex and FSBIRecord nodes. Macroeconomic data, including the Leading Economic Index (LEI), Personal Consumption Expenditures (PCE), and Consumer Price Index (CPI)—were retrieved from FRED and stored in the graph as Indicator nodes.

Given the limitations of running Neo4j locally on Colab, we migrated the graph to Neo4j AuraDB, enabling secure, cloud-based storage with persistent Bolt protocol connectivity. Environment variables were configured in Colab for secure credential management, and the pipeline was refactored to work seamlessly with AuraDB, allowing us to query the graph and extract features for modeling directly within the notebook environment.

We implemented a scraper to collect Yahoo Finance news articles for each company. Texts were preprocessed and analyzed using the FinBERT transformer, generating sentiment scores (positive, negative, neutral) for each article. These scores were aggregated monthly and stored as properties on Document nodes linked to their respective companies.

Features extracted from the knowledge graph included: monthly stock returns (target variable), FSBI metrics (sales index, sales_mom_pct), FRED indicators (LEI, PCE, CPI), aggregated FinBERT sentiment scores, and lagged stock returns (up to 3 months) and moving averages. An XGBoost regression model was trained with an 80/20 time-based train-test split. Model evaluation included Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) per company, along with overall model performance metrics (accuracy, F1 score, ROC AUC). Feature importance was analyzed to understand the contribution of each variable to the model.

Our ingestion workflow combines manual curation with automated pipelines:

- Manual curation: We maintain a CSV file of 10 consumer-discretionary companies (symbol, name, sector) and a JSON file defining product categories and subcategories.

Company	% Wtg
Amazon.com, Inc.	21.3%
Tesla, Inc.	16.0%
The Home Depot, Inc.	7.3%
Booking Holdings Inc.	4.8%
McDonald's Corporation	4.8%
The TJX Companies, Inc.	4.1%
Lowe's Companies, Inc.	3.6%

Starbucks Corporation	2.6%
O'Reilly Automotive, Inc.	2.3%
Chipotle Mexican Grill, Inc.	2.0%

Figure 1 - The Consumer Discretionary Select Sector SPDR Fund (XLY): https://finance.yahoo.com/quote/XLY/holdings/

With respect to the market data, daily returns were downloaded from the module yfinance and stored as StockReturn nodes. Sector index metrics from Fiserv's Small Business Index (FSBI) were downloaded separately as CSV exports containing sales and transaction volume levels, along with month-over-month and year-over-year percent changes. A module called the FSBILoader reads and normalizes these records into FSBIIndex and FSBIRecord nodes linked to their industries. Finally, macroeconomic context is incorporated by fetching time series specifically for the Leading Economic Index (LEI), Personal Consumption Expenditures (PCE), and the Consumer Price Index (CPI) from January 1, 2022, onward via the FRED REST API. In future work, the SEC dataset can be pulled from the Edgar API. Risk attributes as well as key financial ratios (debt-to-equity, ROA, etc.) can be derived from parsed SEC statements and attached to quarterly filing nodes. Each series is stored as an Indicator node with associated child records capturing observations by date.

All raw inputs (HTML, JSON, and CSV) are archived under data/raw, then passed through a lightweight parsing layer that converts them into Python dictionaries before feeding parameterized Cypher statements to Neo4j over the Bolt driver. This multi-modal pipeline produces a richly connected graph spanning textual descriptions, corporate fundamentals, sector signals, and economic time series, ready for downstream competitive-intelligence analysis.

Database Schema

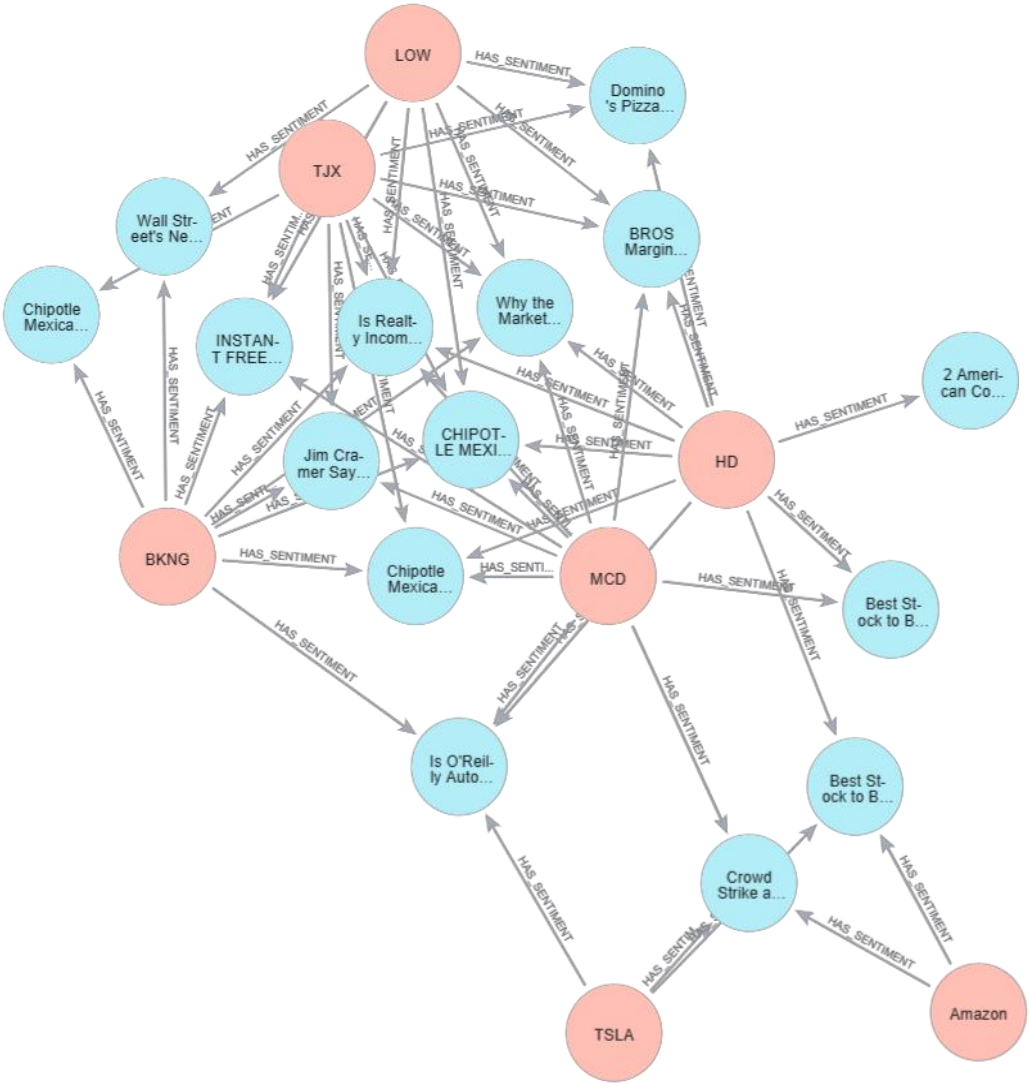

Figure 2 - Neo4j Aura DB schema

The knowledge graph employs a rich schema of node labels, properties, and relationships to capture corporate, product, index, and macroeconomic entities.

Nodes in the graph are typed according to the following labels, each with a specific set of properties:

- Company: symbol, name, CEO, headquarters, marketCap, wikiURL, displayLabel, embedding, id, tags
- Document: id, text, source, tags, date, displayLabel, embedding
- FSBIIndex: index_id, seriesName (as name), valuesByDate (stored under value arrays), embedding, id, date
- FSBIRecord: sales_index, sales_mom_pct, sales_yoy_pct, transaction_index, trans_mom_pct, trans_yoy_pct, date, id
- Indicator: indicator_id, name, valuesByDate (value arrays), embedding, id, date
- IndicatorRecord: value, date, id, type
- Industry: name, displayLabel, id
- Person: name, role, id, displayLabel
- Product: name, displayLabel, id
- StockReturn: date, value, id, displayLabel
- Sentiment: date, score, source

Collectively, property keys across all nodes include: date, displayLabel, embedding, id, index_id, indicator_id, name, role, sales_index, sales_mom_pct, sales_yoy_pct, symbol, tags, text, trans_mom_pct, trans_yoy_pct, transaction_index, type, and value.

Entities are interconnected through typed relationships that encode business context:

- PART_OF_INDUSTRY (Company → Industry)

- OFFERS / PRODUCTS (Company → Product)
- COMPETES_WITH (Company ↔ Company)
- SUBSIDIARY_OF (Company → Company)
- HAS_EMPLOYEE (Company → Person)
- OF_COMPANY (Person or Document → Company)
- OF_INDEX (FSBIRecord → FSBIIndex)
- OF_INDICATOR (IndicatorRecord → Indicator)
- AFFECTS_INDUSTRY (Indicator → Industry)
- HAS_SENTIMENT (Indicator → Company)

This schema allows storage of static attributes (e.g., company headquarters), time-series metrics (stock market returns, FSBI and FRED records), sentiment-based metrics and multi-hop traversals across corporate, product, and economic dimensions.

The current knowledge graph has been populated with 11 company entities, 14 industry categories, 81 distinct products, 134 source documents, 10 FSBI index series, and the 3 macroeconomic indicators from FRED and 9,274 relationships. This level of coverage reflects the integration of corporate, product, news sentiment and economic signal data into a unified graph structure.

Results

Overall, the results were very strong however not necessarily surprising. Considering the data were gathered over a 30-day period during which the market experienced historical daily gains, the autoregressive nature of the model demonstrated a highly accurate predictor of future daily gains. In other market environments with longer

durations that include differing business cycles, the sentiment media measure should provide a greater lift to the market prediction capabilities.

The aggregated results across the 10 companies in our sector highlighted a strong model:

Mean model.n_estimators = 100 — indicates that our XGBoost model used 100 boosting rounds (trees), as expected.

Accuracy = 0.903 ± 0.017 — our model correctly classifies (or predicts the direction of) stock returns about 90% of the time, with low variability across folds.

F1 Score = 0.907 ± 0.016 — the model balances precision and recall well, meaning it's effective at distinguishing between up and down market days.

ROC AUC = 0.969 ± 0.010 — This indicates that our model is highly effective at separating positive vs. negative classes.

The modeling across all 10 stocks can be found below:

	MAE Mean	MAE Std	RMSE Mean	RMSE Std
AMZN	0.293985	0.007767	0.353084	0.010719
BKNG	0.298214	0.017152	0.365232	0.017726
CMG	0.297844	0.024310	0.358795	0.022288
HD	0.304659	0.028735	0.367035	0.037393
LOW	0.309585	0.021821	0.373777	0.029022
MCD	0.297394	0.016342	0.366997	0.010040
ORLY	0.291993	0.030461	0.361293	0.033679
SBUX	0.293267	0.010724	0.355809	0.017372
TJX	0.304846	0.031651	0.362766	0.034151
TSLA	0.297381	0.024638	0.360231	0.027583

Mean Absolute Error (MAE): On average, the model's prediction is off by about 0.29 to 0.31 — suggesting reasonable precision in predicting the magnitude of returns (given returns usually range between -0.2 to +0.2).

Root Mean Squared Error (RMSE): Measures the spread of errors — typically 0.35 to 0.37 — indicates our model is consistent in performance across tickers with some natural variation.

Conclusions

The Neo4j knowledge graph effectively models companies, industries, products, corporate hierarchies, business metrics, and macroeconomic indicators. Its flexible schema and Cypher-driven queries facilitate answers to strategic questions relevant to competitive intelligence. Management can use this platform to identify overlapping product offerings, map competitor networks, analyze subsidiary structures, and integrate business metrics with economic trends.

This project demonstrates that a Neo4j-based knowledge graph, extended with FinBERT sentiment analysis and deployed via AuraDB in Colab, provides a powerful platform for competitive intelligence and predictive modeling in the consumer discretionary sector. The XGBoost model's strong overall performance—accuracy of 0.903, F1 score of 0.907, and ROC AUC of 0.969—demonstrates the potential of knowledge graph-driven features. However, feature importance analysis highlighted that lagged stock prices were the primary drivers of predictive accuracy, while sentiment features contributed meaningfully but with lower relative importance. Future work could expand the knowledge graph to additional companies, incorporate real-time sentiment streams, and apply advanced hyperparameter tuning and ensemble methods to refine model predictions.