Regime Prediction in The Stock Market:

(https://github.com/srijanrodo/erdos_regime_switch)

Overview: Regime detection has been a crucial field of investigation ever since the market crash in 2009. Broadly speaking, the stock market oscillates between two states: **Bearish** and **Bullish**, signified by a period of great volatility and steady growth, respectively. In this project, we aim to predict the aforementioned regimes by examining the trading data of stocks for several companies in various sectors. Additionally, we incorporate sentiment analysis in the previous analysis and investigate possible correlations.

Stakeholders: Effective regime prediction models have been used by traders, financial institutions, portfolio managers, and government agencies. It's an active area of research including Economics, Social Sciences, and Operations Research. Some papers have been attached below that inspired the analysis in this project.

Data Gathering: We obtain historical data for the stocks: Apple ('AAPL'), Google/Alphabet ('GOOGL'), Nvidia ('NVDA'), Delta Airlines ('DAL'), Verizon ('VZ'), Exxon Mobil ('XOM'), Chevron ('CVX') from stooq.com. Furthermore, we focus on AAPL and scrape relevant news headlines from businessinsider.com to perform sentiment analysis.

Data Preprocessing:

- Based on existing research on this subject, we begin our analysis with volatility, normalized daily returns, daily high, and daily low as our features.
- We apply Kalman filters to smooth noise in raw market data and ease model convergence. This serves as the training data.
- To train models, we use Gaussian Hidden Markov Models to calculate regime-states corresponding to every timestamp(day) and stock.
- For sentiment analysis, we first use a pre-trained LLM and clustering to remove duplicates of the same news. Next, we use simple-regex to remove headlines that are relatively unrelated to the stock being

considered. Finally, we use FinBERT to generate weighted sentiment scores for each headline and average them over each day.

Modeling Approaches: We train and select the best models for each individual stock separately. At the outermost layer, we treat the problem as a time-series problem.
Once we select a time series split for cross-validation, we train traditional regression models using a sliding window into the past to predict the current day market state and restrict the window size to at most 25 to follow the Markov assumption during training.
We also vary the window size to find the optimal point for each stock. Here are the models considered:

- k-Nearest Neighbor
- DecisionTree
- AdaBoost
- XGBoost
- RandomForests
- Support Vector Machines *
- Logistic Regression *
- Constant baseline prediction

Due to class imbalance in both training and testing data, as the market tends to be more bullish than bearish on average, we use both accuracy and f1 scores to compare performance.

Results: For individual stocks, the best results across all models and window sizes through cross-validation are attained by either LogisticRegression or SVM. We run the corresponding models for each stock over the test time-series dataset and record performance. Specifically, the latest time-series data between the dates of Aug 3rd 2023 and Dec 31 2024 is reserved for testing. Model performance is described in the table below:

Stocks	AAPL	GOOGL	NVDA	DAL	XOM	CVX	VZ
F1 (bullish)	0.98	0.94	0.85	0.99	0.97	0.95	0.95
F1 (bearish)	0.70	0.62	0.73	0.88	0.64	0.59	0.73
Accuracy	96%	90%	80%	99%	95%	92%	92%

Remarks: The exceptional performance on DAL can be attributed to greater class imbalance on collected data. The relatively poor performance on NVDA can be attributed to the training data being collected before the LLM boom of 2023 and the testing data spanning the period of 2023-2025 when the stock surged compared to its previous behaviour until 2022.

Future Directions: Here are some of the possible extensions we can look into in the future:

- Implement robust sentiment analysis from social media/more news sources for other stocks.
- Choose more stocks and create models that cover multiple stocks/segments of the market at once.
- Price prediction using the best-performing supervised learning algorithms. Implement a backtesting engine for live trading simulations.
- Consider more than two states and sophisticated metrics for volatility computation to represent more volatile/unpredictable periods in the market and capture large-scale behaviour along with its impact on particular assets.

Literary Sources: The following list is by no means an exhaustive list of papers in regime detection, but primary sources of inspiration for this project.

- An, Sufang, et al. “Early warning of regime switching in a complex financial system from a spillover network dynamic perspective.” iScience, vol. 28, no. 3, 2025.
- Wang, Matthew, et al. “Regime-Switching Factor Investing with Hidden Markov Models.” J. Risk Financial Management, vol. 13, no. 12, 2020.

- Franke, Jürgen. “Markov Switching Time Series Models.” Handbook of Statistics, vol. 30, 2012, pp. 99-122.

Additional References: The following articles were especially enlightening towards this project’s success.

- Luck, Spencer. “Time Series Regime Analysis in Python.” medium.com, 13 Oct 2022,
- Holls-Moore, Michael. “Hidden Markov Models for Regime Detection using R”.