Regime Switch Prediction

Data Science Boot Camp
(Spring 2025)

Project By: Suman Bhandari, Srijan Ghosh, Arka Karmakar, Shubham Saha, Sridhar Venkatesh

Overview

- Model the stock market with two states: Bearish and Bullish
- Use time series stock data to predict states in the future
- Look at individual stocks to represent different segments of the market
- Consider sentiment analysis as an additional feature

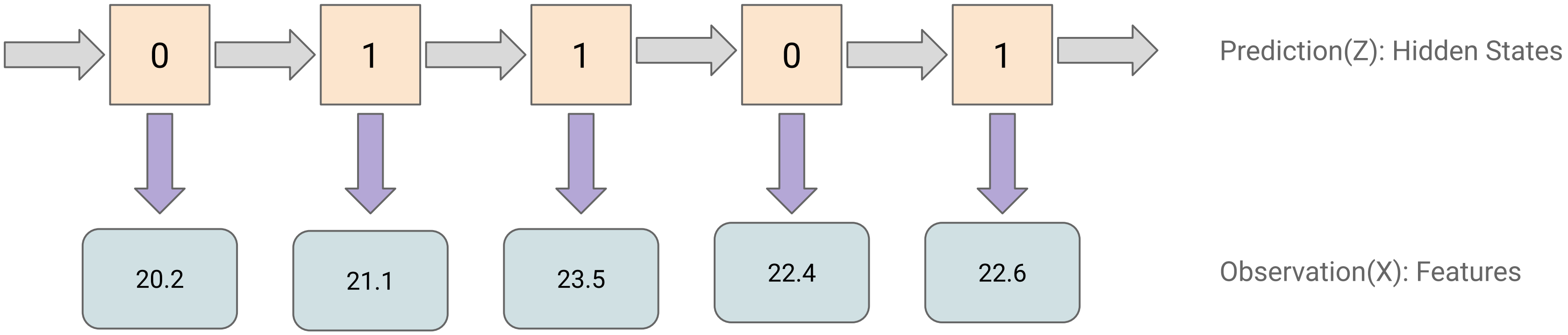
Data Sources:

- stooq.com
- businessinsider.com

References:

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

Gaussian Hidden Markov Model To Generate Labels



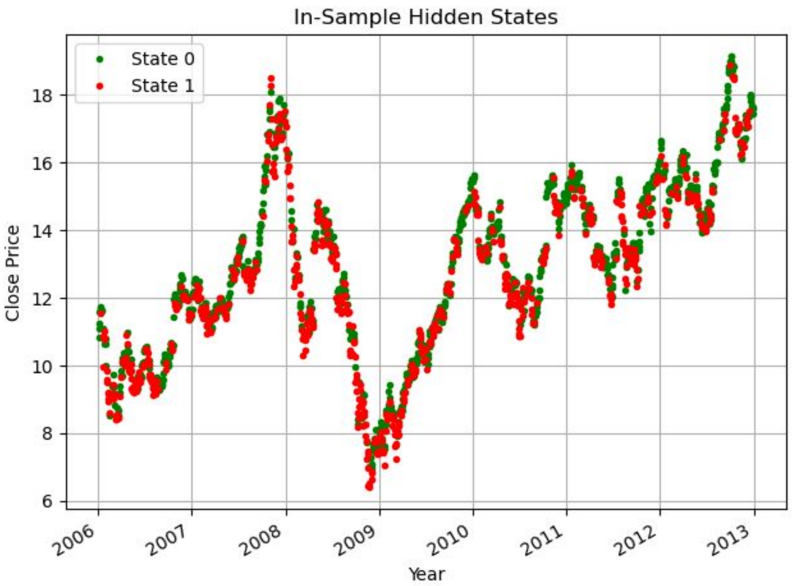
Features we use as observations:

- Daily Returns
- Daily High Price
- Daily Low Price
- Volatility

Gaussian HMM: X is given by a Gaussian distribution conditioned over Z.

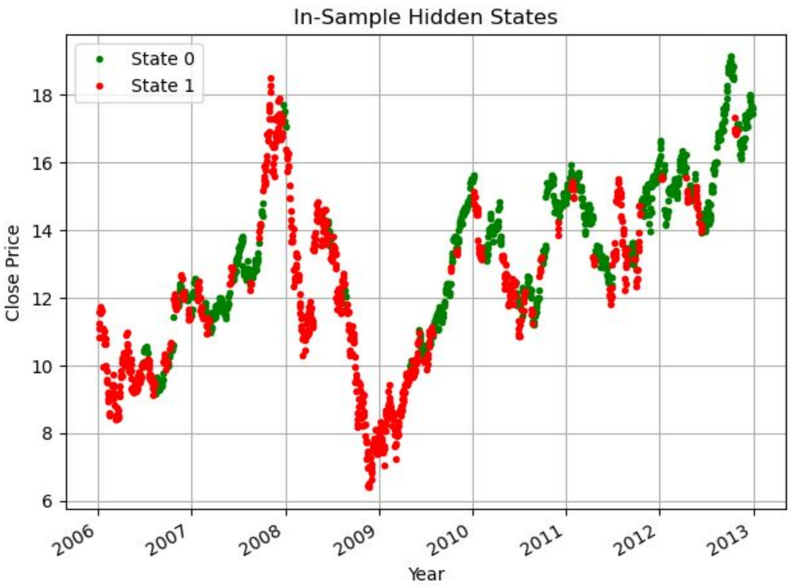
Processing Features before HMM

Issue: Observed features are noisy



Labeling: Without Kalman filters

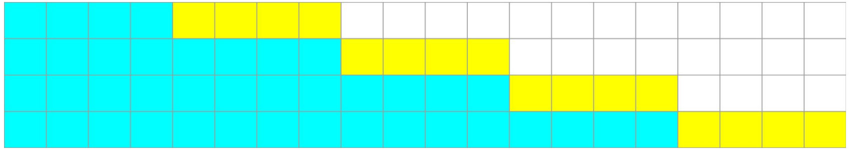
Solution: Smoothen with Kalman Filters

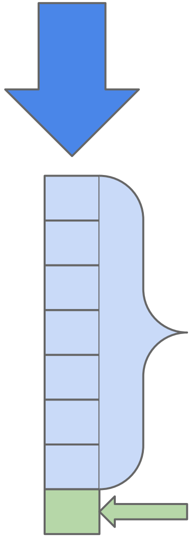


Labeling: After applying Kalman filters

Plots: “GOOGL” prices (2006-2013)

Training with Cross-Validation





Observations over *n*-days as features

State of the *(n+1)*th day as target

TimeSeriesSplit for cross-validation

For each time series, create sliding windows of size *n*

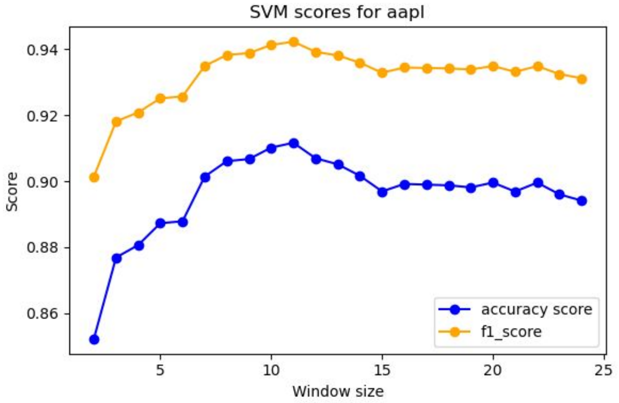
Optimize for *n* to find best windows over different models and stocks

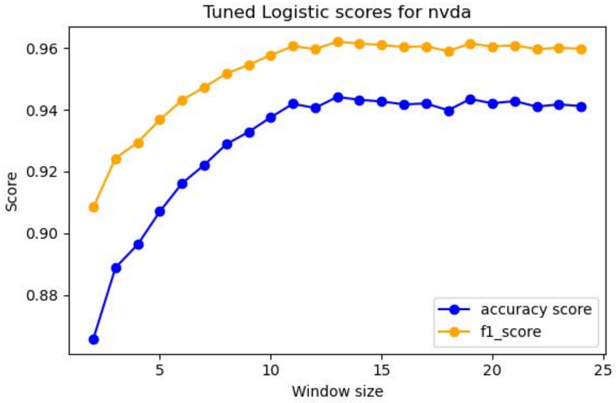
Training Results

Models:
- k-Nearest Neighbor
- Decision Tree
- AdaBoost
- XGBoost
- Random Forests
- Support Vector Machines
- Logistic Regression
- Constant baseline prediction

Metrics:
- Accuracy Score
- F1 Score

For each model we optimize over the size of the sliding window and use optuna/gridsearchcv to tune hyperparameters





Testing Results

Final Testing Data (unused for training): The period Aug 3rd, 2023 to Dec 31st, 2024

Best Models: Support Vector Classifier for AAPL and VZ, Logistic Regression for all the others.

Final Results for Each Stock

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:

- DAL has class imbalance resulting in high accuracy.
- NVDA has relatively low accuracy owing to the recent LLM boom and corresponding volatility in NVDA stock.

Addendums

Some more results:

- Using sentiment analysis on news headlines and using it as an additional feature does not lead to a notable improvement over the same window size.
- We used the model trained on "GOOGL" data to predict states for MSFT with f1_score 0.97 (bullish) and accuracy 95%. This suggests that the models perform well on stocks within a similar sector.

Future Directions:

- Implement robust sentiment analysis.
- 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.

Thank You!