Exploring Unsupervised Learning for Stock Returns Forecasting

Ismail Berbache Alessandro Morosini Ellie Yang Michelle Zhou Steve Zhou

All code provided here.

December 9, 2024

Contents

1 Introduction

1.1 Context and Purpose

Financial markets are complex systems influenced by numerous interconnected factors and non-linear dynamics. Predicting stock prices requires methods that uncover patterns without overfitting to noise. Traditional models like ARIMA are limited by linear assumptions, while advanced techniques such as Long Short-Term Memory networks (LSTMs) can model non-linear dependencies.

In this project, we explore the hypothesis that clustering stocks by time series similarity enhances the accuracy of the forecasting models. Using k-means clustering with Dynamic Time Warping (DTW) and Fourier Coefficients as distance metrics, we group FTSE 250 stocks to isolate meaningful patterns within clusters. These will serve as the foundation for both linear and non-linear predictive models. Specifically, we train ARIMA, LightGBM, and LSTM models on clustered data, comparing their performance against models trained on individual stock series or the entire dataset. We aim to determine whether clustering helps filter out irrelevant signals while incorporating valuable contextual data, addressing overfitting and noise.

1.2 Data

For this project, we used historical stock price data from the FTSE 250 index, sourced from Yahoo Finance. The dataset includes daily trading information, such as open, high, low, close, adjusted close prices, and trading volumes over the past 10 years. Our analysis focuses on the adjusted close price, which accounts for corporate actions like dividends and splits, providing a consistent basis for modeling. The FTSE 250, a benchmark index of mid-cap stocks in the UK, offers a diverse cross section of industries, making it ideal for clustering and predictive modeling. The decade-long time span captures varied market conditions, providing a robust foundation for testing our methods.

After loading and verifying the raw FTSE 250 historical stock data, we performed critical preprocessing to ensure meaningful and statistically sound analyses. We inspected the dataset for missing values, discarding tickers with over 20% gaps and using time-based interpolation and forward/backward filling for the rest, reducing the sample to 169 stocks. Anomalies, including a misreported ticker scaled by a factor of 100, were corrected. We then standardized each time series using rolling means and standard deviations to normalize price fluctuations, ensuring comparability across stocks for clustering. Finally, the cleaned dataset was formatted for clustering and predictive modeling.

2 Time-Series Clustering

2.1 Dynamic Time Wrapping

Dynamic Time Warping (DTW) clustering is a method designed to group time series based on shape similarity, accommodating temporal misalignments by stretching or compressing segments of the series. This flexibility makes DTW especially suited for financial time series, where stocks may follow similar trends but on different timescales due to varying market dynamics. In this project, DTW is used as one of the distance metrics explored within k-means to group FTSE 250 stocks based on their adjusted closing prices from 2017 to 2021. While computationally intensive, this method reveals nuanced relationships between time series, enabling the construction of forecasting models that incorporate insights from similar stocks.

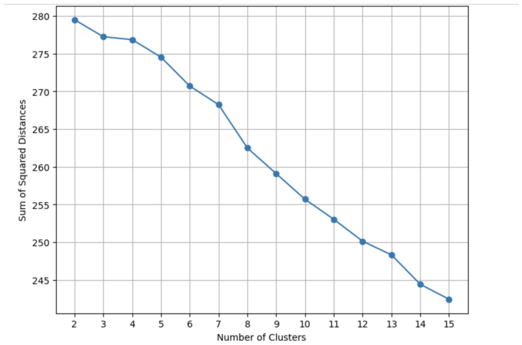

Figure 1: Scree Plot of DTW K-Means

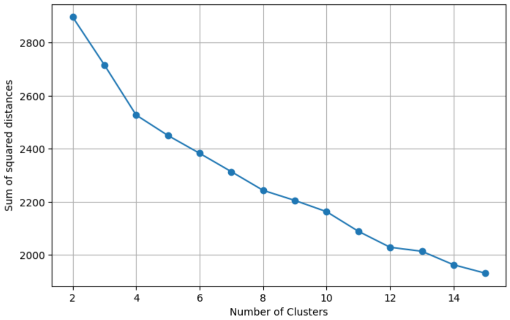

Figure 2: Scree Plot of Fourier Coef K-Means

2.2 Fourier Coefficients

Since the stock price is a time series, we can leverage Fast Fourier Transformation to extract the most important frequencies and their corresponding magnitudes for each stock. These Fourier coefficients encapsulate periodic patterns that are of the greatest intensities in the time series. While using all fitted coefficients gives a precise reconstruction of the time series, we selected only the top-n frequencies with the greatest magnitudes (measured by the norm of the complex magnitudes) to avoid a high dimensionality in clustering. We set n to 10 upon a manual comparison of the reconstructed series and the original data.

We fitted two types of clustering, starting with K-means on the Fourier coefficients. We used the scree plot to choose $K = 8$. In addition, we recognize that K-means assumes feature alignment, but stocks can exhibit similarity in non-aligned frequency ranks, e.g., one stock's top-1 frequency might align with another's top-2. Thus, we also computed stock similarity using the magnitude-weighted frequency differences to the closest matches and used this to create an Agglomerative clustering.

3 Forecasting Leveraging Clustered Stocks

In this section, we evaluate the performance of various models for stock price prediction by training on data from January 2017 to December 2021 and testing on data from January 2022 to December 2022. Each model is trained and evaluated across three configurations: global data (combining all stocks), clustered data (grouping stocks into `n_cluster = 8` clusters), and individual stock data. For individual stock training, for each stock, the model is trained and evaluated using only its historical data. For clustered stock training, the model is trained on all data within a cluster and tested individually for each stock. For all stock training, all stocks are combined into a single dataset for training and then tested on individual stocks.

Model performance is assessed using two metrics: Root Mean Square Error (RMSE), which measures the average magnitude of prediction errors, and Mean Absolute Percentage Error (MAPE), which quantifies prediction accuracy as a percentage of actual values.

3.1 ARIMA

The baseline model we use is ARIMA(AutoRegressive Integrated Moving Average), a widely used statistical method for time series forecasting. As ARIMA is designed for univariate analysis, we only train the model on individual stocks but not on global and clusters.

3.2 LightGBM

The first model we used in our analysis is LightGBM, a gradient-boosting framework based on decision trees. LightGBM is designed for high efficiency and scalability, making it particularly well-suited for large datasets and complex tasks such as predicting stock price.

Feature Engineering & Model Configurations

We performed feature engineering to enhance the dataset by generating lag features, rolling statistics, and technical indicators. Specifically, we included lagged values of stock prices, moving averages, rolling standard deviations, and extreme values (max/min) over different window sizes(5,10 and 20). We also calculated the rate of change (ROC) and Relative Strength Index (RSI) for each stock.

We utilized GridSearchCV to tune the hyperparameters, and use this set of hyperparameter to train all the future LightGBM models: feature_fraction': 0.9, 'learning_rate': 0.05, 'n_estimators': 300, 'num_leaves': 15, 'reg_alpha': 0.1, 'reg_lambda': 0.1.

3.3 LSTM

The second model we trained on to explore the performance of model training on clustered stocks verses individual or all stocks is LSTM, the Long Short-Term Memory networks.

Feature Engineering & Model Configurations

To make sure the LSTM learns temporal dependencies and patterns effectively, we converted our stock data into a windowed format and then treated each window as an independent training example. Each window consists of 3 historical data points used to predict the next stock price.

The architecture of the LSTM model comprises two stacked LSTM layers with 256 and 128 units, followed by two dense layers with 64 and 32 units using ReLU activation, and a final output layer with a single neuron for regression. The model is compiled with the Mean Squared Error (MSE) loss function and optimized using the Adam optimizer with a learning rate of 0.001.

4 Result & Discussion

The baseline model of ARIMA on individual stocks have the result of rmse 191.46 and mape 24.85%. Then we trained the models using data from individual stocks, all stocks combined, and stocks within the same cluster as the target stock. For LightGBM, we evaluated six different clustering results derived from various clustering methods and reported their performance as in Table 1.

	By Individual	By Global	By Cluster (DTW)		By Cluster (FFT)			
Metric	**Individual**	**Global**	**DTW_5**	**DTW_8**	**kmean_5**	**kmean_8**	**hier_5**	**hier_8**
RMSE	34.39	17.23	17.89	21.86	19.82	20.79	18.42	18.49
MAPE	3.99%	1.76%	1.79%	1.97%	1.87%	1.97%	1.83%	1.85%

Table 1: LightGBM Model performance comparison

The results of the best experiments of LightGBM and LSTM are summarized in Table 2 and 3.

Training Configuration	**Average RMSE**	**Average MAPE (%)**
Individual Stock Training	34.39	3.99
Clustered Stock Training	17.89	1.79
All Stock Training	17.23	1.76

Table 2: LightGBM Best Performance

Training Configuration	**Average RMSE**	**Average MAPE (%)**
Individual Stock Training	49.08	3.80
Clustered Stock Training	45.45	2.86
All Stock Training	27.33	2.57

Table 3: LSTM Performance

Our results indicate that using clustered data generally outperforms training on individual stocks but shows no significant improvement over training on all data. This can be attributed to the relatively small dataset size (162 stocks) and the limited noise within the dataset, which reduces the theoretical advantage of clustering. When comparing LightGBM and LSTM, we observe that the boosted tree model performs slightly better. This advantage is primarily due to LightGBM's fast execution speed, which allows it to handle more features and larger window sizes effectively. LSTM still have great potential for improvement with access to more computational resources.

5 Future Work

To build upon this work, we propose the following directions for future research. We can expand the dataset to include more stocks, enabling a deeper evaluation of clustering's advantages in larger and noisier datasets. Additionally, we aim to explore a wider range of models and incorporate higher-dimensional features, such as market indicators and macroeconomic data, to assess whether clustering in a richer feature space enhances performance. Finally, designing and testing trading strategies based on model predictions will allow us to evaluate the practical utility of the models and determine if clustering yields tangible benefits in real-world trading applications.

References

[1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794. ACM, August 2016.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[3] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3149–3157, Red Hook, NY, USA, 2017. Curran Associates Inc.

[4] Pavel Senin. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 855(1-23):40, 2008.