

Amrita Vishwa Vidyapeetham

Centre for Excellence in Computational Engineering and Networking

Amrita School of Engineering, Coimbatore

**Dynamic Mode Decomposition for Financial Trading Strategies**

**&**

**(DMD) for Foreground Detection in Video**

**Prepared By Team 11:**

J.R.S. Ajay Surya

V. Mohitha

M. Prasanna Teja

P. Sai Ravula

**Supervised by:**

Dr. Abhijith M S

Asst. Professor

An End Semester Project submitted to the CEN department as a part of course evaluations of **“21MAT301-Mathematics for Intelligent Systems 5”** for

B. tech in **Computer Science Engineering** – **Artificial Intelligence.**

**ACKNOWLEDGMENT**

We are deeply thankful to **Centre for Excellence in Computational Engineering and Networking** (CEN) at **Amrita Vishwa Vidyapeetham**, **Coimbatore** for providing us such a wonderful environment to peruse ourresearch. We would like to express our sincere gratitude to **Dr. Abhijith M S, Asst. Professor, Department of Centre for Excellence in CEN**,Amrita Vishwa Vidyapeetham. We have completed our research under hisguidance. We found the research area, topic, and problem with his suggestions. He guided us with our study and supplied us many research papers and academic resources in this area. He is patient and responsible. When we had questions and needed his help, he would always find time to meet and discuss with us no matter how busy he was. We deeply appreciate your contributions to our team’s journey.

We would also like to extend our gratitude to **Dr. K P Soman**, **Head of the Department, School of Artificial Intelligence**, at CEN, Amrita Vishwa Vidyapeetham, who provided us with the base knowledge of this field of research, without him this research work would not exist in its present form. We would also like to acknowledge our team members for supporting each other and be grateful to our university for providing this opportunity for us.

**Individual contributions**

**Team 11**

****

**TABLE OF CONTENTS**

**DMD applications**

* 1. – Abstract . . . . . . . . . . . . . . . . . . . . . . . .
  2. – DMD for Financial Trading strategies . . . . . . . . . . . . . . . . . . . . .
  3. – Problem Description
  4. – Methodology
  5. – Algorithm
  6. – Flow chart
  7. – Results
  8. – DMD for Foreground Detection in Video. . . . . . . . . . . . . . . . . . .
  9. – Problem Description
  10. – Methodology
  11. – Implementation
  12. – Results

4 – Conclusion . . . . . . . . . . . . . . . . . . . . . . .

5 – References . . . . . . . . . . . . . . . . . . . . . . .

**Abstract**

This report presents two applications of Dynamic mode Decomposition (DMD). The first one is Dynamic Mode Decomposition for Financial Trading Strategies and the other one is DMD for foreground Detection in video. Algorithmic trading schemes are growing of importance in modern financial world. Each year, increasing proportion of the total trading volume is handled by algorithmic trading systems and they have become a fundamental element of modern day trading. We propose a method for price prediction using Dynamic Mode Decomposition assuming stock market as a dynamic system. The method is capable of characterizing complex dynamical systems, in this case financial market dynamics, in an equation free manner by decomposing the state of the system into low-rank terms whose temporal coefficients in time are known. We have used the modes for the predictive assessment of the stock market. We also used this concept to recommend the stocks based on growing eigenvalues by creating a GUI using streamlit.

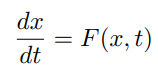
**Dynamic Mode Decomposition for Financial Trading Strategies**

**2.1 Problem Description**

Stock price prediction is a challenging problem as the market is quite unpredictable. Changing dynamics of the financial system adds to the non-stationarity of the data which makes long term predictions inaccurate. Also time series analysis is computationally costly. In our proposed method we assume stock market to be a dynamical system and use DMD (Dynamic Mode Decomposition), a data driven, spatial-temporal coherent algorithm to identify the evolutionary patterns of this system. DMD is computationally very efficient as it exploits the low dimensional structure of the data. A dynamical system can be decomposed to modes. These modes help us determine how the system evolves and the future state of the system can be predicted. We have used these modes for the predictive assessment of the stock market. Since DMD modes are dynamic modes, they capture the trend of market in them . We worked with the time series data of the companies listed in National Stock Exchange. The granularity of time was minute. We have sampled a few companies across sectors listed in National Stock Exchange and used the minute-wise stock price to predict their price in next few minutes.

**2.2 Methodology**

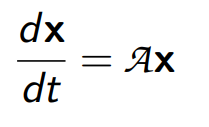
We consider financial system as a non-linear dynamic system whose governing equations are not known to us. The snapshots we take corresponds to a state of the system. Snapshot of a system consists of observed measurements of that system at time t. In our case the observed measurements are the stock price of each company at the time t.

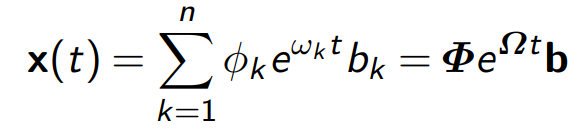
 A dynamic system is described using a governing set of differential equations.

At each state we can make different kinds of measurements of the observables. The measurement function can be denoted as

where k = 1, 2,...,M where M is measurement time.

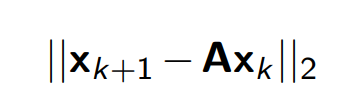
The initial condition is stated as

The non-linear function F that defines the set of governing equations is unknown. All we have is the initial conditions and the measurements taken . In our case the measurement is the stock price. In DMD procedure, we construct an approximate linear evolution of the system.

The solution for the above equation is

Here ψk and ωk are the eigenvectors and eigenvalues of matrix A.

**If real part of eigenvalues are positive and greater than one, then it means a growing mode and growing money. If eigenvalues are negative, then it means decaying modes and losing money**

The ultimate goal in the DMD algorithm is to optimally construct the matrix A so that the true and approximate solution remain optimally close in a least-square sense

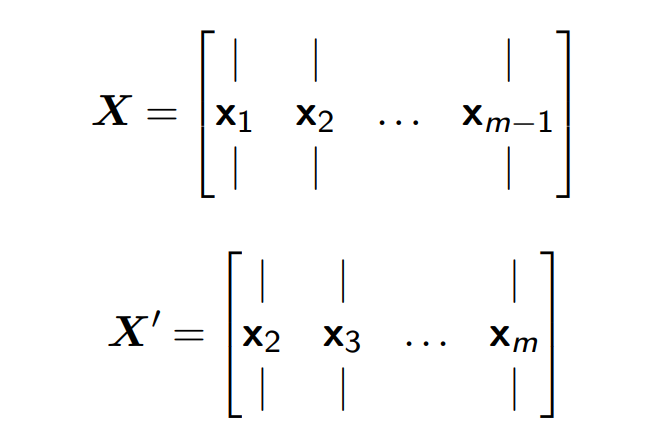
**2.3 Algorithm**

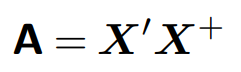
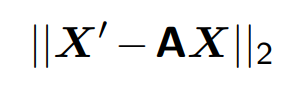
The data collection process involves two parameters:

N = number of companies in a given portfolio

M = number of data snapshots taken

The data snapshots are arranged into an N × M matrix

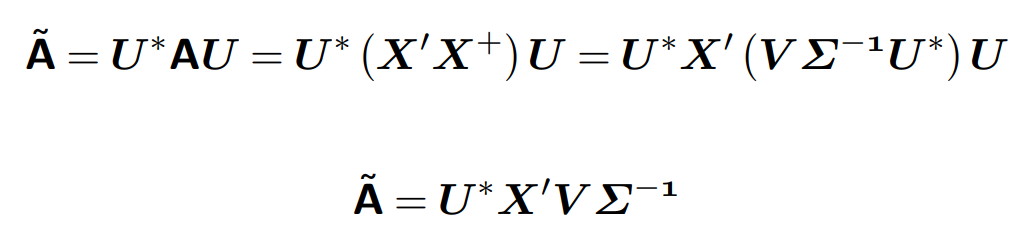
**Step I :** From this matrix we need to capture the underlying dynamics of the system. For this we splits the data matrix into

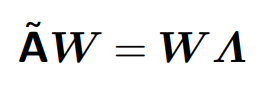
* The locally linear approximation may be written in terms of the data matrices
* The best approximate matrix A is given by the Moore Penrose pseudo-inverse of X
* This solution minimizes the error:

**Step II :** Find the truncated SVD of the data matrix X, to obtain U, Σ and V matrices, such that

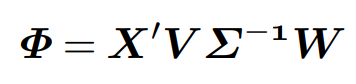
where;

* U ∈ Cn×r , Σ ∈ Cr×r , and V ∈ Cr×m
* Here ‘r’ is the rank of the reduced SVD approximation to X
* The left singular vectors U are POD modes
* The columns of U are orthonormal, so U∗U = I and V ∗V = I

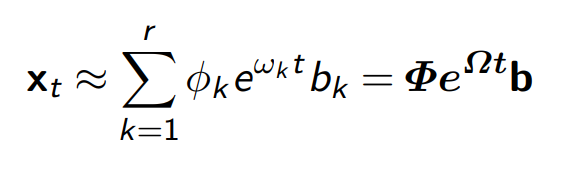
**Step III :** Find the reduced (‘r × r’) projection of the full matrix A onto the orthogonal POD modes (U), given by (A˜) as;

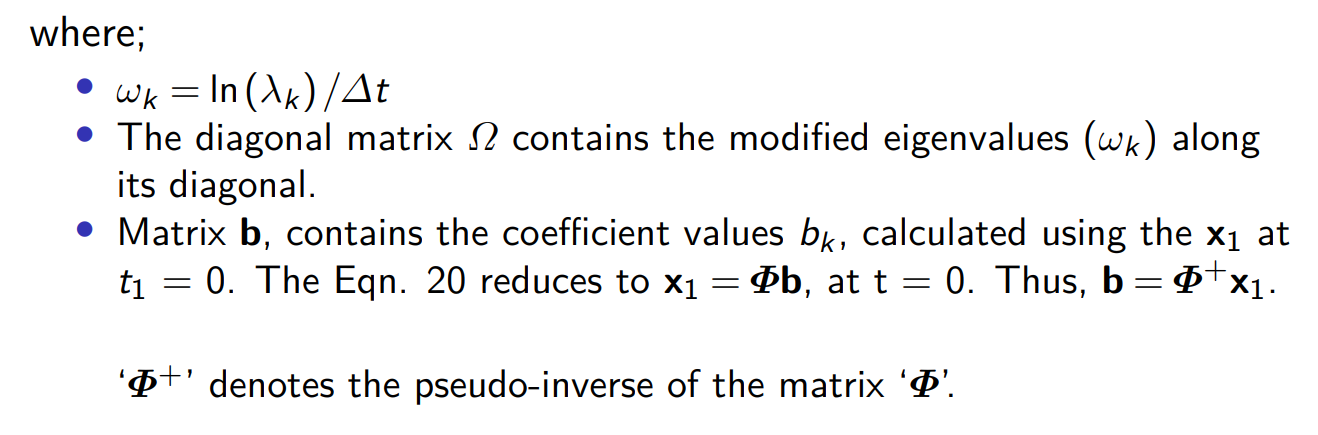
**Step IV :** Compute the eigenvectors and the corresponding eigenvalues (λk ) of A˜, such that

where the columns of the matrix W are the eigenvectors of A˜ and ‘Λ’ is the diagonal matrix containing the corresponding eigenvalues λk .

**Step V :** Finally, the eigenvectors of the full matrix A are obtained using W. The eigenvectors of A (DMD modes) are given by columns of Φ;

‘Φ’ (called exact DMD modes) contains the exact eigen vectors of the full matrix A and ‘Λ’ contains all the non-zero eigenvalues of the full matrix A

* The calculated DMD modes (Φ) and eigenvalues Λ are used to predict the future states of the system. The approximate solution at all future times is given by

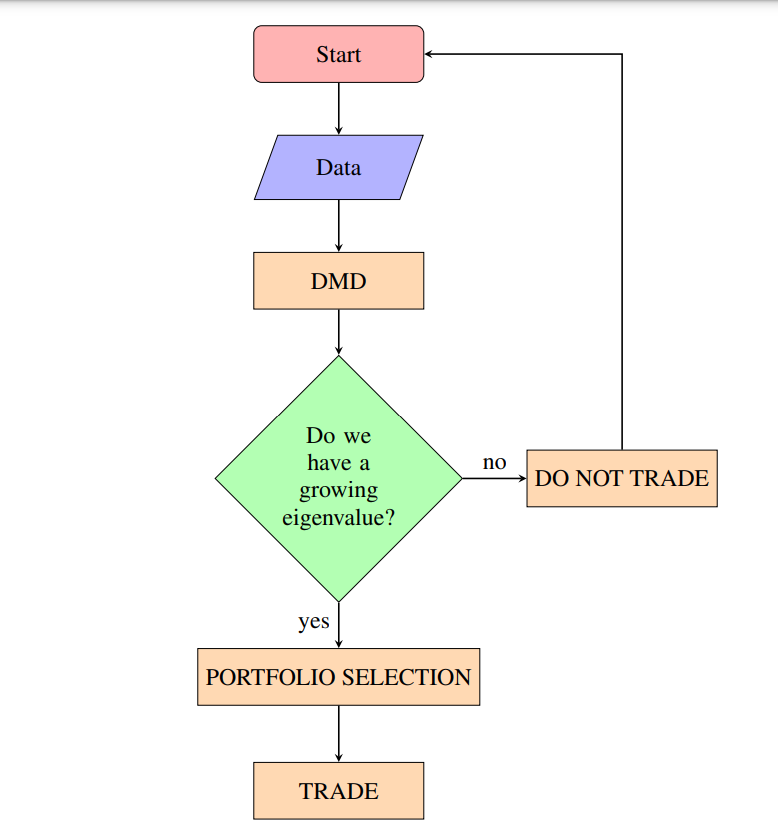


**Rank Mismatch Problem**

* when data has fewer measurement points than time points
* n represents measurement points, the number of rows of the data and m represents time point, the number of columns of the data.
* When n < m then the SVD step of the DMD process produces at most n singular values and this restricts the number of DMD modes and eigenvalues to n. The number of DMD modes and eigenvalues may be insufficient to capture the dynamics over m snapshots in time.
* If we try to construct the DMD solution of a financial index because of the large difference between the number of rows and the columns of the data matrix, we will have a rank mismatch issue

**2.4 Flowchart**

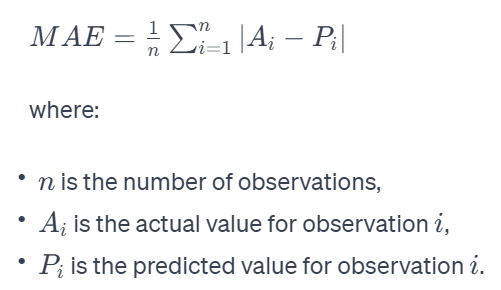
The real parts of the DMD eigenvalues give the growth rate and the complex parts give the frequency of the oscillation. For our purposes, we do not use the complex part of the eigenvalues, we only focus on the real parts.

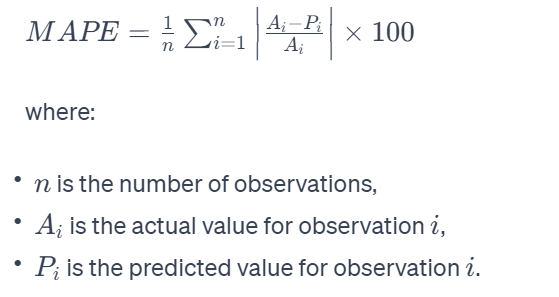
The real part of the DMD eigenvalue represents the growth rate of the associated DMD mode. Since we do not interested with their complex part, we can separate them into two groups: The first group is the eigenvalues which have growth rate larger than one and the second group is the eigenvalues which have growth rate smaller than one. We are interested with the first group, we call them **growing eigenvalues.**

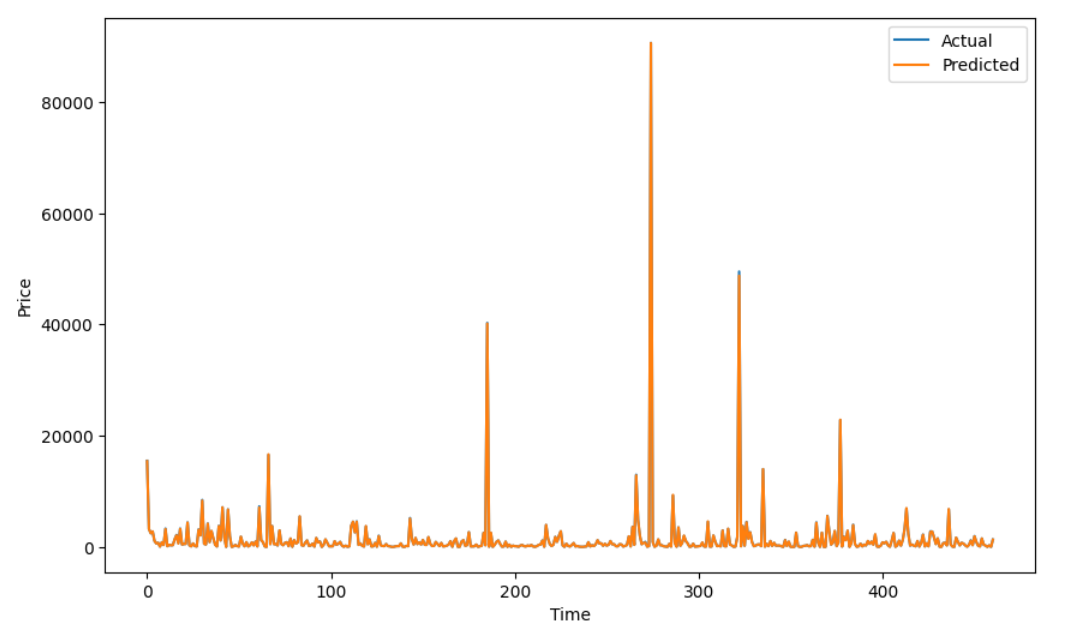
**2.5 Results**

After loading the dataset NSE500i and converting it to a dataframe ,we splitted the data for training and testing. DMD method from Pydmd library is imported and the dataset is fitted. Finally model is evaluated using Mean Absolute Error (MAE) between the predicted values and actual values . We got MAE value as 15.5299.. and a Mean Absolute percentage Error (MAPE) of 1.94% .In practical terms, a MAPE of 1.94% suggests that the model's predictions, on average, have an error of less than 2% relative to the actual values. This level of accuracy is often considered quite good, especially in forecasting and prediction tasks.

The formula for MAE is as follows:

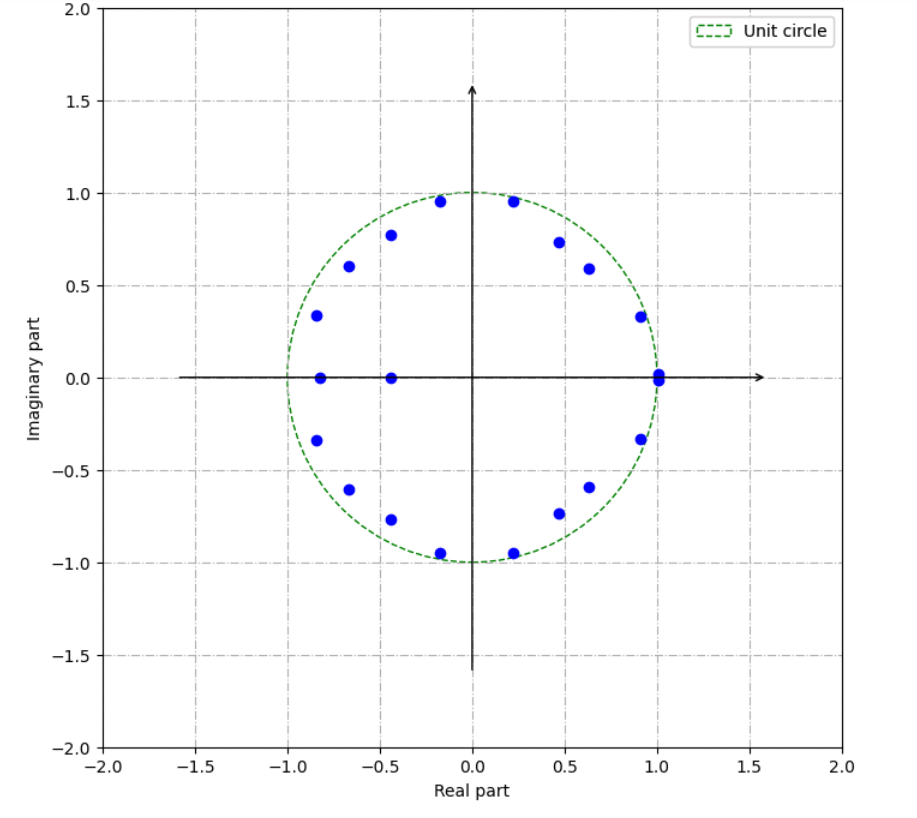


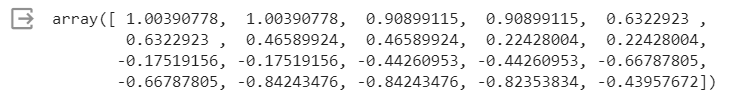
For a given set of predictions and corresponding actual values, the MAPE is calculated as follows:

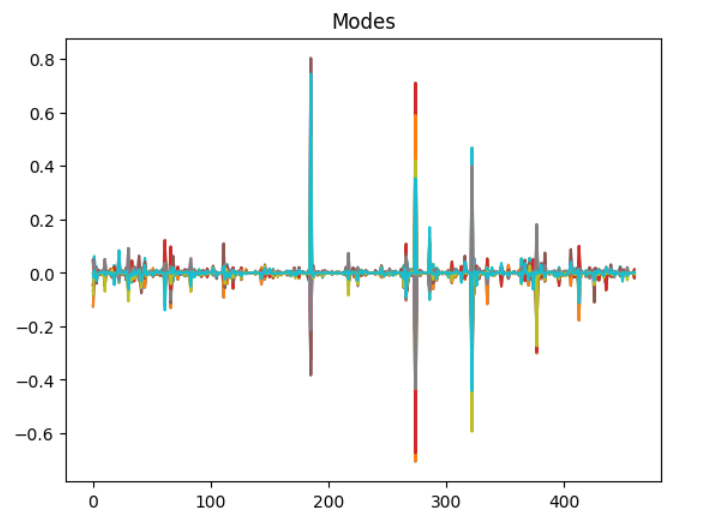
The plot showing the values of actual and predicted prices is shown below

Eigen value analysis: We iterated through each eigenvalue obtained from the DMD analysis (dmd.eigs).

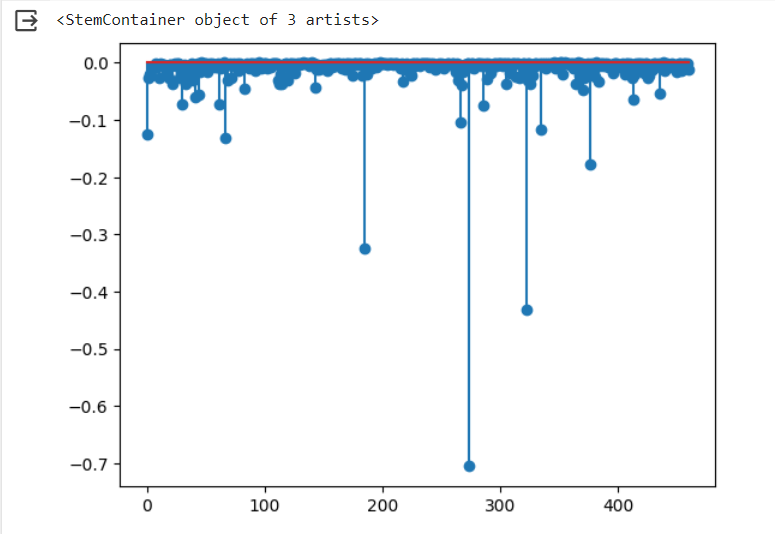
For each eigenvalue, you calculate its distance from the unit circle using the Euclidean distance formula. The distance is printed along with the eigenvalue

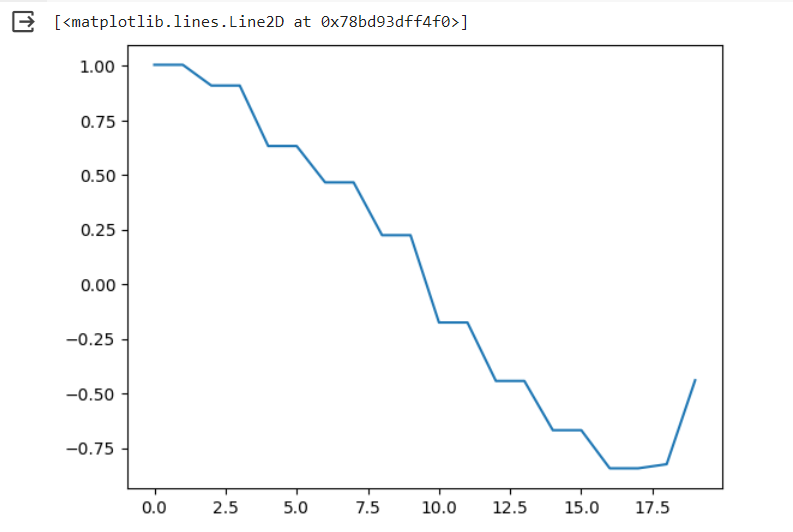


 Next iterating through each mode obtained from a Dynamic Mode Decomposition (DMD) analysis and plotting the real part of each mode on the same graph. We also retrieved the real parts of the eigenvalues using **dmd.growth\_rate**

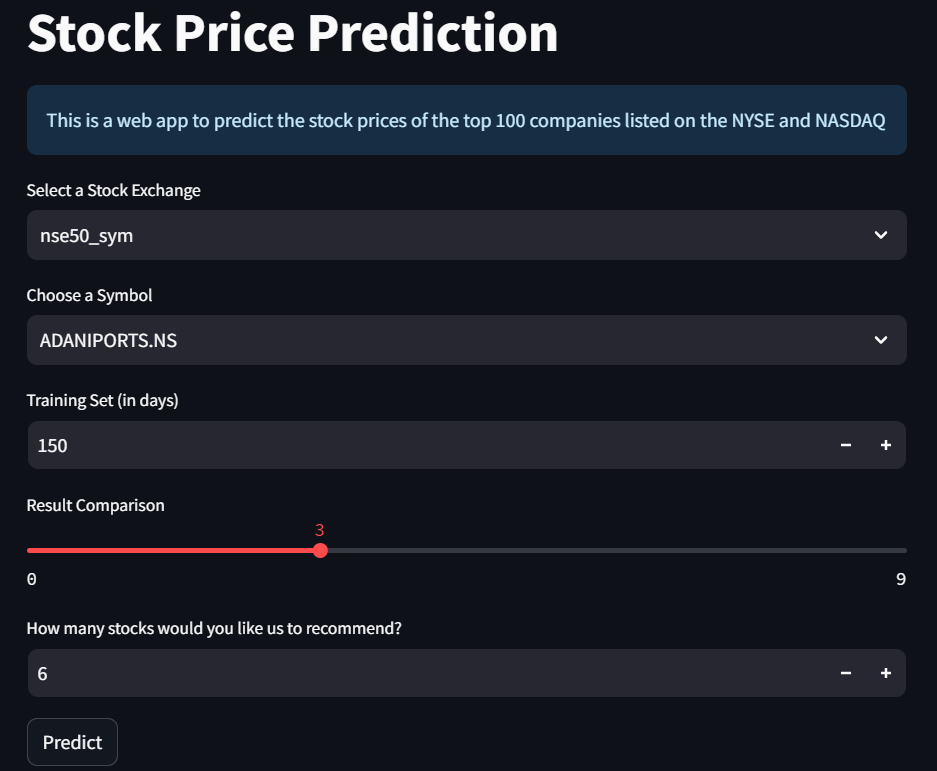


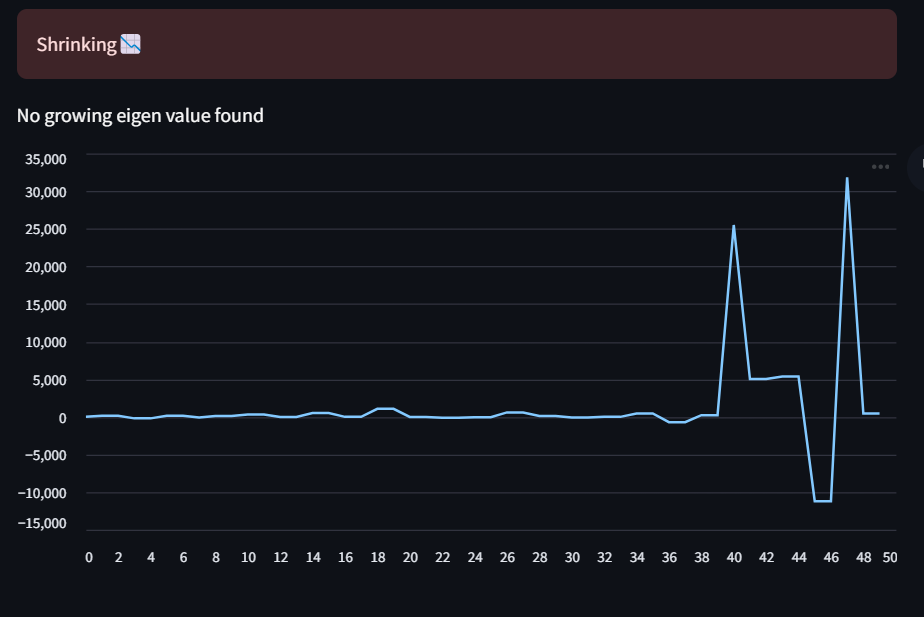
Using **plt.stem** to create a stem plot of the real part of a specific mode obtained from a Dynamic Mode Decomposition (DMD) analysis . This stem plot can be helpful in gaining a visual understanding of the temporal behavior and characteristics of the selected mode. You may observe patterns, oscillations, or trends in the plot, which can provide insights into the dynamic behavior captured by the DMD analysis



The below plot will show the magnitudes of the growth rates associated with each mode. Positive values indicate growth, negative values indicate decay, and the magnitude represents the rate of growth or decay.

Below picture depicts the Streamlit based GUI developed for Stock recommendations and predictions.

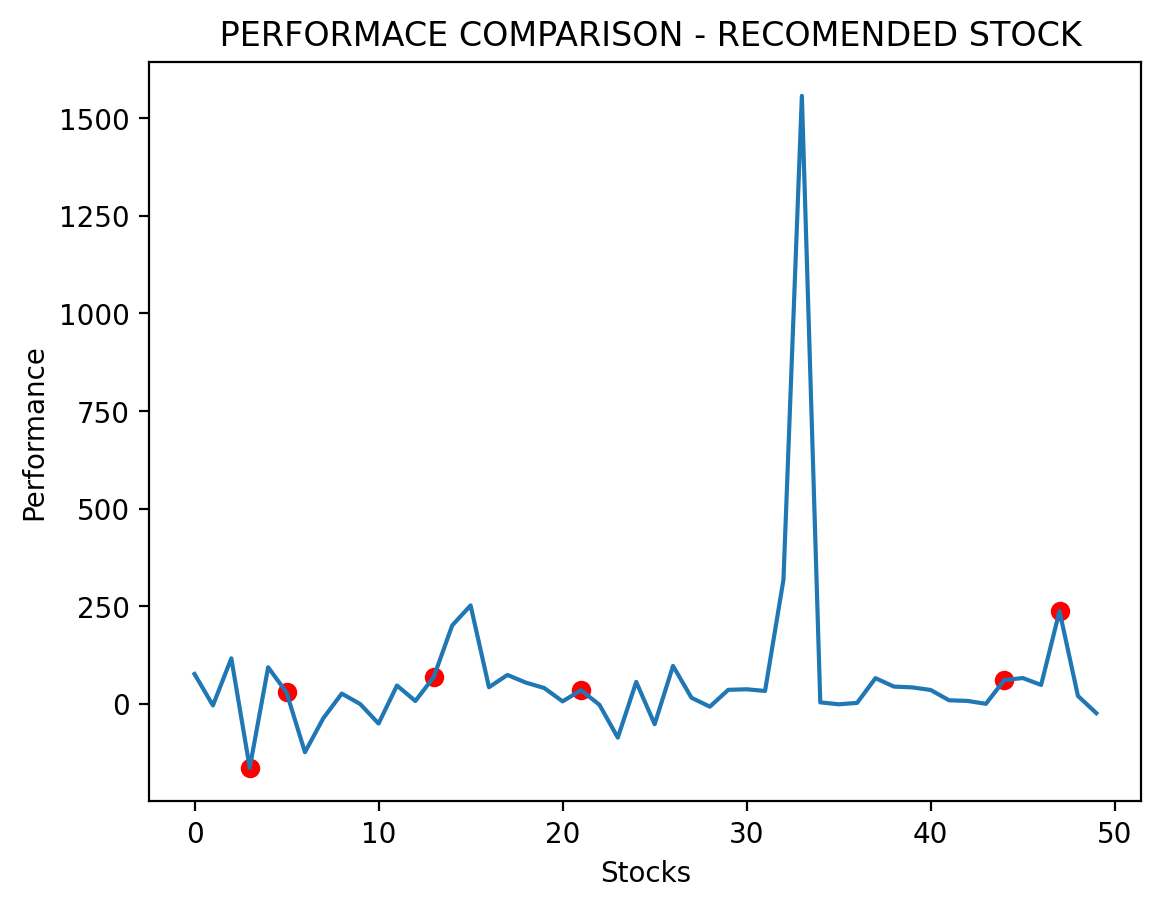


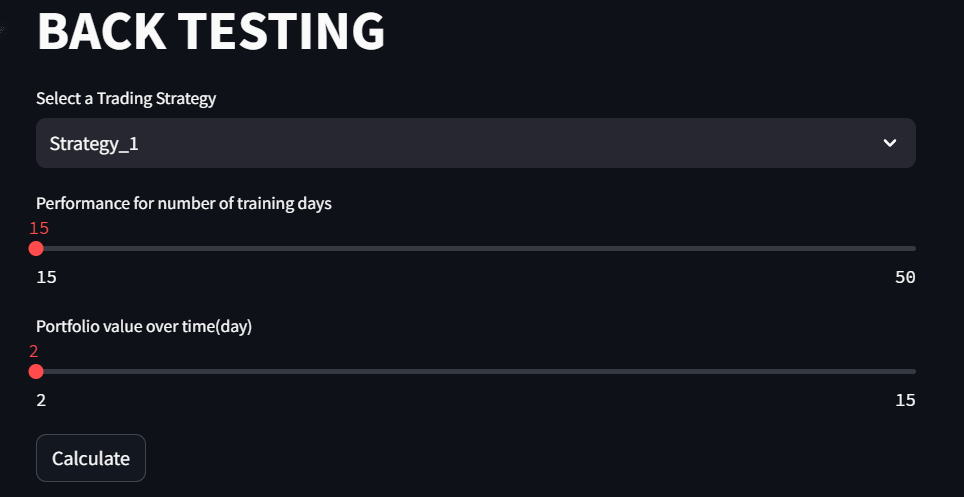


We will get recommendations based on the analysis done by the model as shown.

A black background with white text

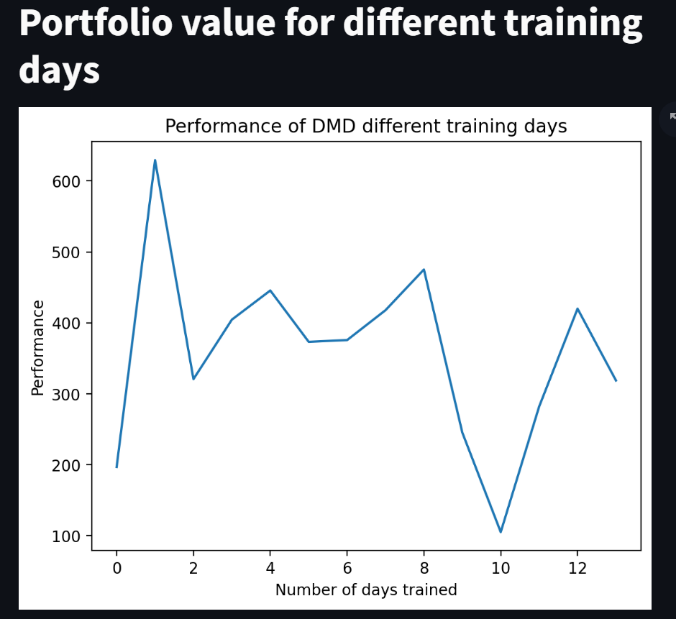
Description automatically generated



Below is the Streamlit GUI for the Portfolio selection after performing the analysis of stocks of different companies.

After running the above GUI, we get the analysis done by the DMD model under different training days.

A graph on a black background

Description automatically generated

**Dynamic Mode Decomposition For**

**Background/Foreground Separation in a video**

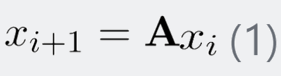
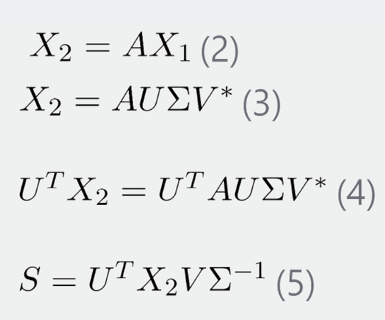
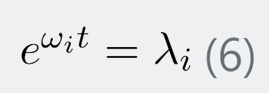
**3.1 Problem Description**

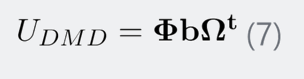
* Imagine you have a video stream capturing the motion of a man in a street. The objective is to use DMD to distinguish between the relatively stable background and the dynamic foreground.
* Unlike traditional methods like SVD and PCA, DMD would capture modes corresponding to the oscillatory behavior associated with the dynamic elements

**3.2 Methodology**

The DMD procedure broadly operates on the following mechanism:

* Given that some collected data is organized into frames in time, each measurement can be organized into a column and acted upon by an operator A.

* This ‘A’ matrix takes the vectorized data from one point in time xi to next point in time xi+1  following eq(1).
* In our approach, rather than finding A directly, dimensionality reduction allows to take only the most dominant (low-rank) modes of the system, and capture their time dynamics, for background subtraction.
* Therefore, we look for a representation of X2 (which contains all of the xi+1 columns stacked side by side ) in the basis of U from the SVD of X1.
* ****As such, the time operator that was used is S , which is defined as UTAU. Given the definition of SVD (i.e. X=UΣV\*, S follows from Equations 2 through 5.
* Thus, by taking the eigen decomposition of S, our DMD modes can be constructed
* Since A and S are related by a similarity transform (unitary rotation & dilation), they share eigenvalues and eigenvectors.
* The DMD modes Φ are then given from the projection of these eigenvectors, y onto the original SVD basis, via Φi = Uyi.
* ****For considering the oscillatory nature of the data, the eigenvalues λ, are related to complex exponentials, via Equation 6, which can be rewritten in terms of ω
* It is from these terms that the DMD time-dynamics are contained, and thus determines how each mode is affected in time. Then, the dynamics of the basis mode, U, is given by the sum over all basis states, or

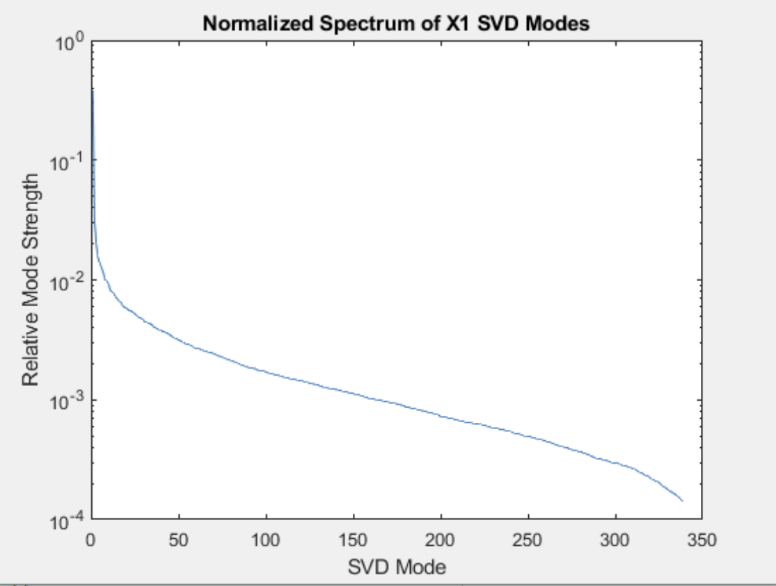
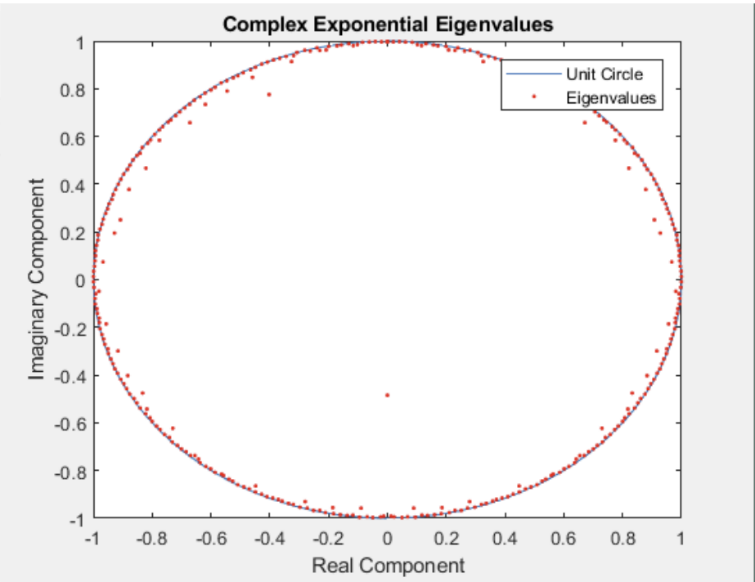


where Ωt is a diagonal matrix of the eωit terms and b comes from the boundary condition determined by Φ-1ui where ui is the first SVD basis mode.

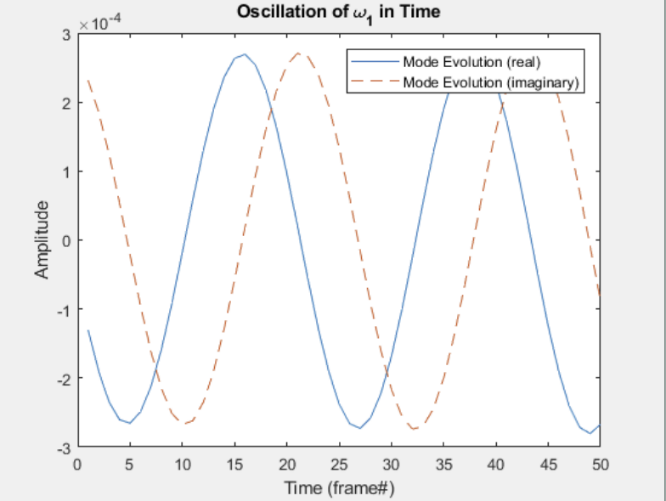
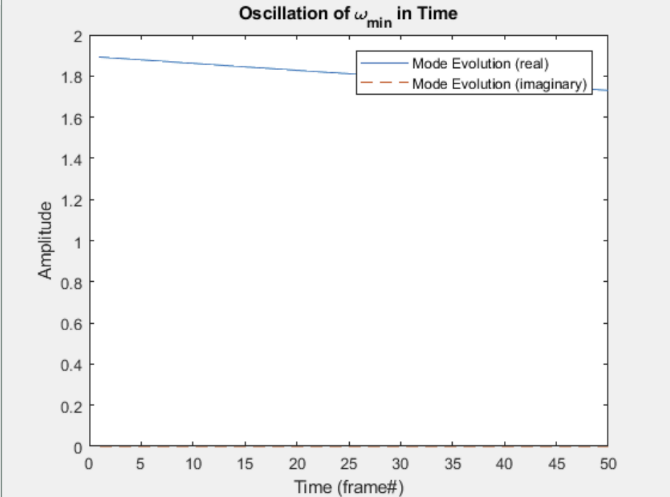
* Finally, the background separation comes from the fact that the low rank modes (background) have small changes in time, and thus their Ωt terms are much smaller in magnitude.
* ‘Pulling out’ these low rank modes, the remaining values make up the sparse time varying information in the video.
* Furthermore, only the absolute value of the low-rank DMD modes are taken, to eliminate possible negative pixel values (which are not allowed in pixel space).

**3.3 Implementation**

1. Once imported, the video is appropriately rescaled (to reduce computation time) and columnated.
2. Then, two matrices are constructed from the original, representing the two data matrices of Equation 2.
3. SVD decomposition, along with calculation of the components of S follow exactly from Equation 5. Again, the time dynamics follow Equations 6 and 7 directly, where the timestep is set as 1 (since the space between each frame is identical in time.)
4. Thus, UDMD becomes the 'background' whose absolute value is subtracted from each frame after normalization.
5. Once the original frame is subtracted from the background, it is the 'foreground' which remains, and a filter is applied to extract a logical matrix from the result, to be used as a filter.
6. After the background filters are returned, they can be utilized raw, or multiplied by the video matrix to extract a foreground.

**3.4 Results**

**Fig.1 Fig.2**

* To set the stage for the decomposition, the SVD spectrum of the data matrix is shown on the left in Figure 1.
* It can be seen that majority of the structure is contained in a very low-rank space. In the spectrum above, the first mode contained 38% of the magnitude of all modes, and thusly has much of the spatial information enclosed.
* ****Furthermore, the oscillatory eigenvalues are plotted in Figure 2, showing that most modes are neither damping or growing, since their absolute values are less than 1.

**original Frame**

** Low Rank Background**

**Foreground Detection**



**Foreground Filtered From original**

**Conclusion**

In conclusion, while DMD offers a valuable tool for analyzing dynamic systems, its application to stock price prediction should be approached cautiously. Financial markets are complex and dynamic, and successful prediction requires sophisticated models that account for a wide range of factors. Consider using DMD as part of a broader analysis, and validate its performance against established financial modeling techniques. Additionally, stay informed about advancements in financial modeling and machine learning, as the field is continuously evolving. For video processing needs, the DMD method allows for much more robust background separation than static methods like SVD, given by its ability to include time-resolved oscillatory behavior to 'update' the background as time progresses. This method could be viable for foreground detection in video streams that have non-ideal (i.e. shaky) background motion and could even provide for a method of creating a 'green-screen' from raw video, although in practice a user may want to use something like a gaussian filter to 'smooth out' the foreground filter. Without any knowledge of the underlying mechanics driving the oscillations, the DMD method allows for equation-free dimensionality reduction, which can be useful in many fields considering nonlinear behavior.

**References**

https://forecasters.org/wp-content/uploads/gravity\_forms/7-c6dd08fee7f0065037affb5b74fec20a/2017/07/kuttichira\_deepthi\_ISF2017.pdf

http://etd.lib.metu.edu.tr/upload/12621107/index.pdf

https://neptune.ai/blog/predicting-stock-prices-using-machine-learning