

**Technical Trading Analysis with Clustering and GenAI – Part I**

(Emergence of AI Qualitative with AI Quantitative Data Analysis)

Warren Camp

Mario Lopez-Gomez

## Emergence of AI Qualitative with Quantitative Data Analysis

### *Abstract*

This paper highlights the birth of the merger of quantitative data analysis with AI-based qualitative data analysis to increase an investors' awareness of market conditions. This paper advocates that a better-informed investor is at less risk. Traditional Technical Trading Analysis is the background for this paper. A seven-step method explores the capabilities of applying GenAI to Stock Market Investment strategies using Python. The emphasis is on the applicability of K-Means clustering to create the Cluster's Centroid and for GenAI (OpenAI gpt-4 API) to classify the image. The data used for this paper is time-series closing price data for a given equity, organized into short-term, medium-term, and long-term vectors. Yahoo Finance provided the pricing data from 1997 to 2024. Over 90,000 data points collected from market, sections, and stocks daily close prices. This paper is the first of two. The second paper will explore the topic of Technical Trading Analysis with Deep Classification. The product of the seven steps developed here in Part 1 is the input to Part II for building Technical Trading Analysis models.

*Keywords:* GenAI, Stock Market, Technical Trading Analysis, python, K-Means, OpenAI, quantitative data analysis, qualitative data analysis

## Contents

- Introduction ..... 4
  - Purpose ..... 5
  - Scope..... 6
  - What is Technical Trading Analysis ..... 6
  - What Is GenAI ..... 6
- Methodology..... 7
  - Step 1: Collect Historical Pricing Data..... 8
  - Step 2 Convert into Vectors..... 9
  - Step 3 – Organize the Vectors into ‘N’ clusters. .... 10
  - Step 4 – Calculate the Centroid for each Cluster..... 10
  - Step 5 – Create Technical Trading Graphical Images..... 10
  - Step 6 – Assess Pattern Using GenAI ..... 12
  - Step 7 – Assign Labels to Patterns ..... 16
- Concluding Remarks ..... 17
- References ..... 20
- Biography ..... 21

### Introduction

Technical Trading Analysis with Clustering and GenAI is Part I of a two-part paper to enhance the capabilities of traditional technical trading analysis. This paper delivers a method that improves timeliness, complexity, interpretation, and accuracy over traditional methods. Using the latest AI tools (Generative Artificial Intelligence (“GenAI”), Generative Adversarial Network (“GAN”), and Deep Classification) combined with a Heatmap-style presentation can provide the investor with holistic market condition experience, see Figure 1 – 3x3 Holistic View of Market Conditions. (Note, this is not the sole information source needed to make sound investment decisions, but Technical Trading Analysis is one of the common processes used by investors).

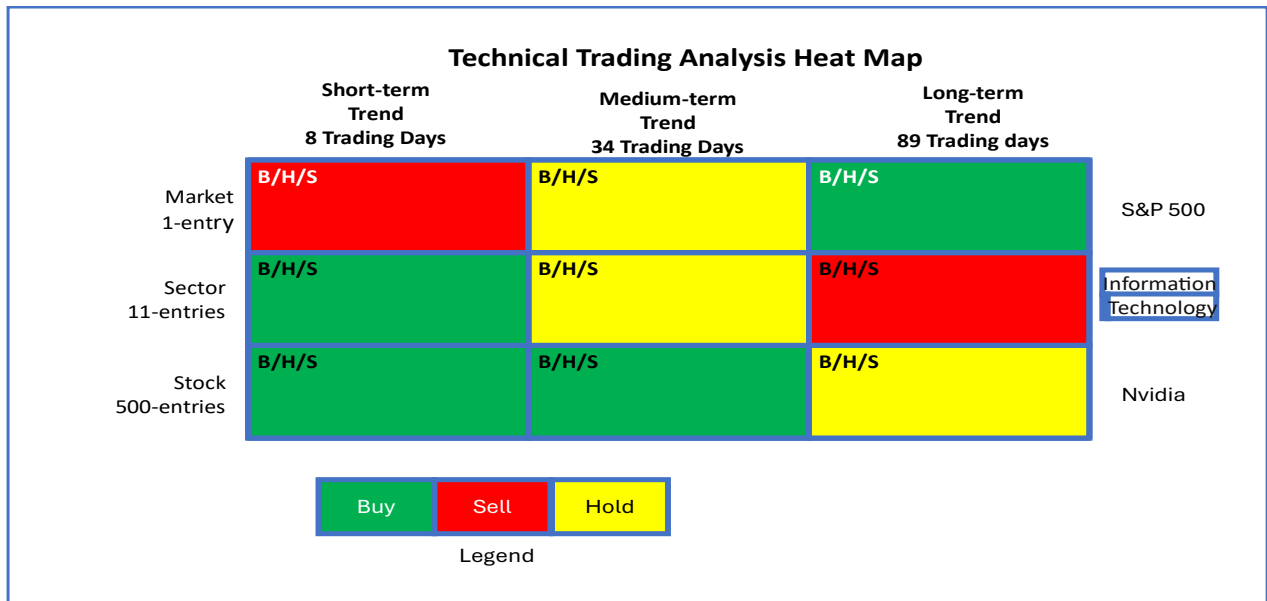


Figure 1 – 3x3 Holistic view of Market Conditions

The paper’s goal is to demonstrate how the latest GenAI tools in text and image processing combined with deep learning classification, can transform traditional Technical Trading Analysis to an automated system capable of efficiently analyzing large volumes of images and identifying high-value, actionable trading patterns for investor consideration, (See Figure 2 for the high-level system architecture.)

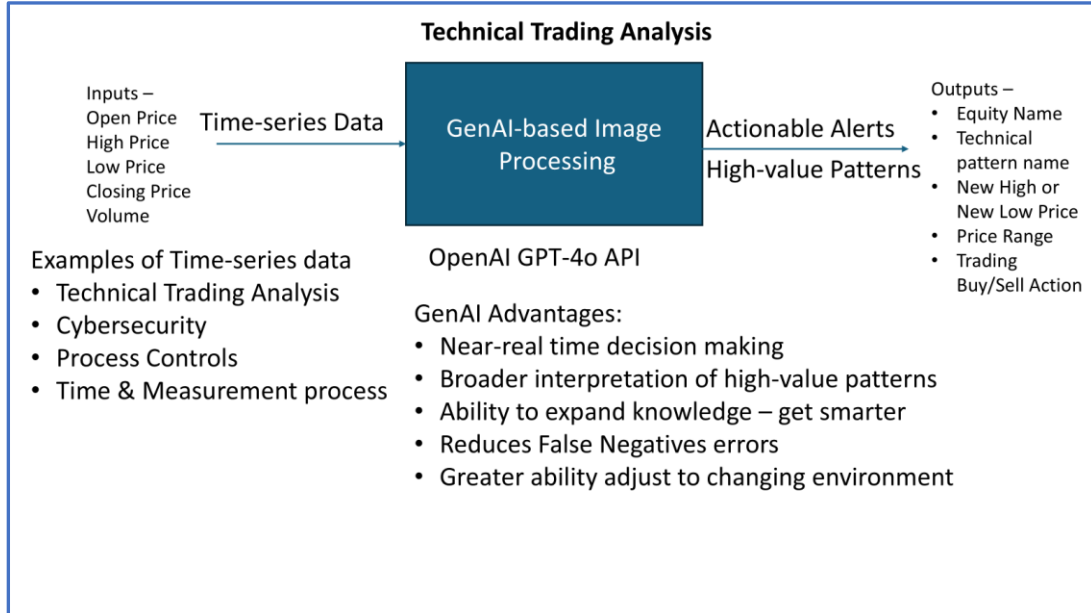


Figure 2- Summary of Proposed Methodology Highlights

In summary, Technical Trading Analysis with Clustering and GenAI can deliver:

- Making near-Realtime decision making.
- Making broader interpretation of high value, actionable trading patterns.
- Expanding investment knowledge (get smarter).
- Reducing Type 3 errors – missing opportunities or bad investments.
- Adjusting to changes in market conditions.

### Purpose

The purpose of this paper is to empower the investor with more timely, accurate, and complete trading information based on Technical Trading Analysis method and the latest AI tools for a more rewarding trading experience.

The paper will show the advantages of a near-autonomous system that can daily analyze hundreds if not thousands of images while filtering the results for high value, actionable trading patterns. This capability has only recently become available in the form of GenAI image processing capability and Large Language Models (LLMs).

### **Scope**

While the combined scope of AI and TTA would be a fascinating topic, it is too broad for the scope of this paper. Our focus will be on long-term (89 trading days) trend analysis using the S&P500 index, SPX, daily closing prices from 1997 to 2024. Other relevant information including price range, new price highs or lows, moving averages, and volume, may help in understanding technical pricing patterns. This implementation currently focuses on a single dimension - the closing price – expandable to two or three dimensions where necessary.

### **What is Technical Trading Analysis**

Technical Trading Analysis is one of the oldest trading methods based on statistical analysis of equity pricing patterns that over time repeat themselves. These patterns usually indicate up or down price movements or points of reflection for price reversals. Patterns are fixed period charts. The charts will highlight key characteristics of the pattern: high- and low-price points, daily price range, and trading volume. Reasons for the popularity of Technical Trading Analysis by traders include Historical Data Utilization, Timing of Trades, Broad Applicability, and Complementary to Fundamental Analysis. While Technical Trading Analysis has a long and successful history, there are opportunities for improvement. Areas of opportunities to make Technical Trading Analysis stronger include avoiding over reliance on past data, understanding false signals, evading complexity, accepting market anomalies, and considering subjectivity in interpretation.

### **What Is GenAI**

GenAI is a revolutionary step in the evolution of Artificial Intelligence. It is revolutionary because it can: 1) understand the meaning of words, and 2) converse in a conversational manner. GenAI uses the latest AI technologies, processes, and data structures none of which individually can describe GenAI. Its true value is the synergy existing in the sum of its parts: for example, parts include GAN, encoding,

## Emergence of AI Qualitative with Quantitative Data Analysis

Natural Language Process (NLP), transformer, large language models (LLM), (See Figure 3 Positioning GenAI and LLMs fit in the AI hierarchy.)

GenAI is the transition from traditional problem solving (quantitative) with specific requirements and decision logic requiring a high degree of control to Prompt Engineering problem solving (qualitative) where the requirements and decision logic are decided by GenAI with lesser degree of control but a more robust solution.

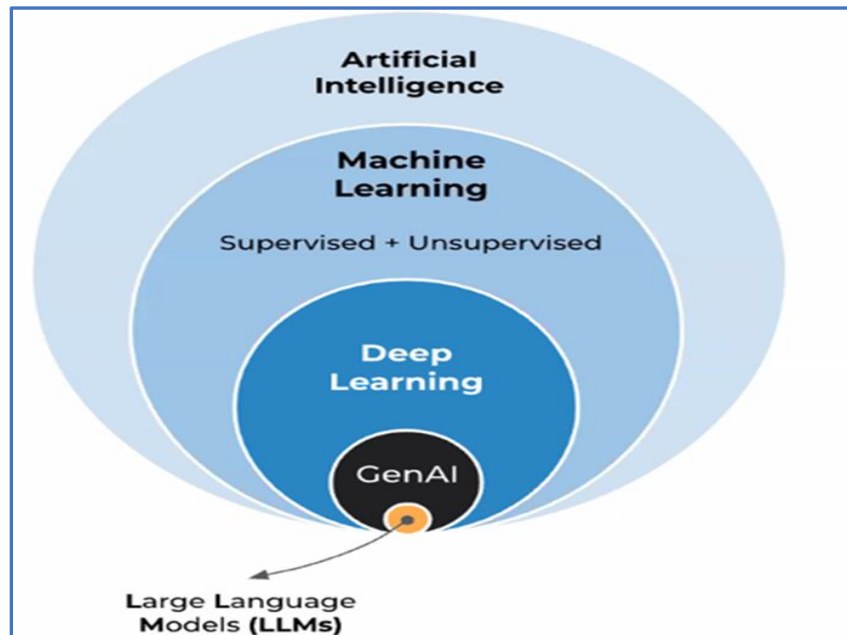


Figure 3 - Positioning GenAI and LLMs in the AI Hierarchy

## Methodology

The proposed method contains seven steps. For each step there is a brief explanation and corresponding python code. Figure 4 displays an overview of these seven steps.

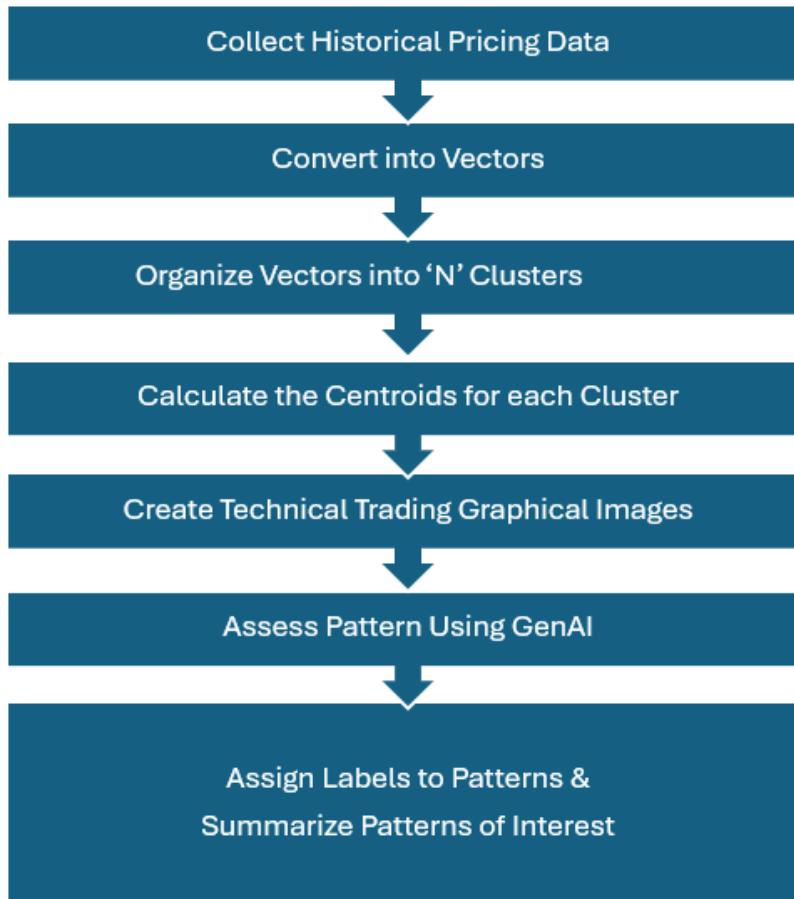


Figure 4 – Overview of the Seven Development Steps.

**Step 1: Collect Historical Pricing Data**

S&P500 Index pricing data downloaded from “for free” websites as CSV files and read by the python program (“See Figure 5 for S&P 500 input data sample.”). Time-series data will organize into vectors whose length corresponds to the number of days in the pattern. Time series data is a sequence of data points indexed in time order.

Stock	TDate	OpenPr	HighPr	LowPr	ClosePr	Volume
SPX	2/23/2023	4019	4028	3969	4012	3952940000
SPX	2/24/2023	3973	3978	3943	3970	3877700000
SPX	2/27/2023	3992	4018	3974	3982	3836950000
SPX	2/28/2023	3977	3998	3969	3970	5043400000
SPX	3/1/2023	3963	3972	3939	3951	4249480000
SPX	3/2/2023	3939	3991	3928	3981	4244900000
SPX	3/3/2023	3998	4048	3995	4046	4084730000
SPX	3/6/2023	4055	4078	4045	4048	4000870000
SPX	3/7/2023	4048	4050	3980	3986	3922500000

Figure 5- SPX (S&amp;P500) Pricing Data

**Python Code:**

```
# Read the CSV file into a DataFrame
df = pd.read_csv(file_path)
df['TDate'] = pd.to_datetime(df['TDate'], format='%m/%d/%Y')
print(df.head(8))
```

**Step 2 Convert into Vectors**

The time-series pricing data converted into 1-dimension vectors, where a vector holds the closing prices for a given length of time; for example, a long-term pattern would contain eighty-nine trading days. Further, it is unlikely that a pattern boundary is the same as the vector, therefore, introduced the concept of overlapping vectors. Overlap refers to time-series data that share common time intervals.

**Python Code:**

```
# Convert df to vectors of length VecLen and overlap VecOL
df = pd.DataFrame(df["ClosePr"])
print(df.head())
# df["ClosePr"] = df["ClosePr"].astype(float)
VecSt = VecLen - VecOL
# Calculate the number of vectors needed
num_vectors = int(np.ceil(len(df) / (VecLen - VecOL)))
print('num_vectors ', num_vectors)
vector = []
vectors = []
# cycle through the input file creating new vectors, the start and stop automatically increment
for each cycle through the data
for i in range(0, num_vectors):
    start_idx = VecSt * i
    end_idx = start_idx + VecLen
    # print(i, start_idx, end_idx)
```

```
vector = df[start_idx:end_idx]
```

### **Step 3 – Organize the Vectors into ‘N’ clusters.**

The next step is to organize Vectors into ‘N’ clusters. For this example, ‘N’ has the value of seventy-five. The approach is to create a cluster of like vectors to stand for similar Technical Trading patterns. While clusters may not hold identifiable patterns, by increasing the number of vectors the number of trading patterns will increase. (Note, we are using Zero-shot Prompting, because the proposed solution only requires one solution description to create the required requirements, considering Few-shot Prompting in a later release.) TimeSeriesK-Means is the cluster method selected. The charted trading patterns are individual cluster’s centroids.

```
Python Code:  
# TimeSeries K-Means clustering with DTW  
model = TimeSeriesKMeans(n_clusters = n_clusters, metric="dtw")  
labels = model.fit_predict(vectors_array)  
model.fit(vectors_array)  
# Get cluster assignments  
cluster_labels = model.predict(vectors_array)
```

### **Step 4 – Calculate the Centroid for each Cluster.**

The cluster’s centroid is the center of the cluster. The K-Means algorithm decides what vectors belong to what clusters. Minimizing distance measurements from the centroid to the vectors, and from the vector to external centroids, results in an iterative calculation until the best centroid location is determined.

```
Python Code:  
# Get cluster centroids  
centroids = model.cluster_centers_  
# The centroid is the center of the Vector Cluster
```

### **Step 5 – Create Technical Trading Graphical Images**

One chart links to one cluster’s centroids. Since the centroid is the center of the cluster, it stands for the similarities of all vectors in the cluster. It is the DNA of the cluster and used to help classify all new vectors.

## Emergence of AI Qualitative with Quantitative Data Analysis

(Note: like most AI algorithms, normalized data requires a range from 0 to 1 or from -1 to 1. Figure 6 is examples of four charts that GenAI will classify.)

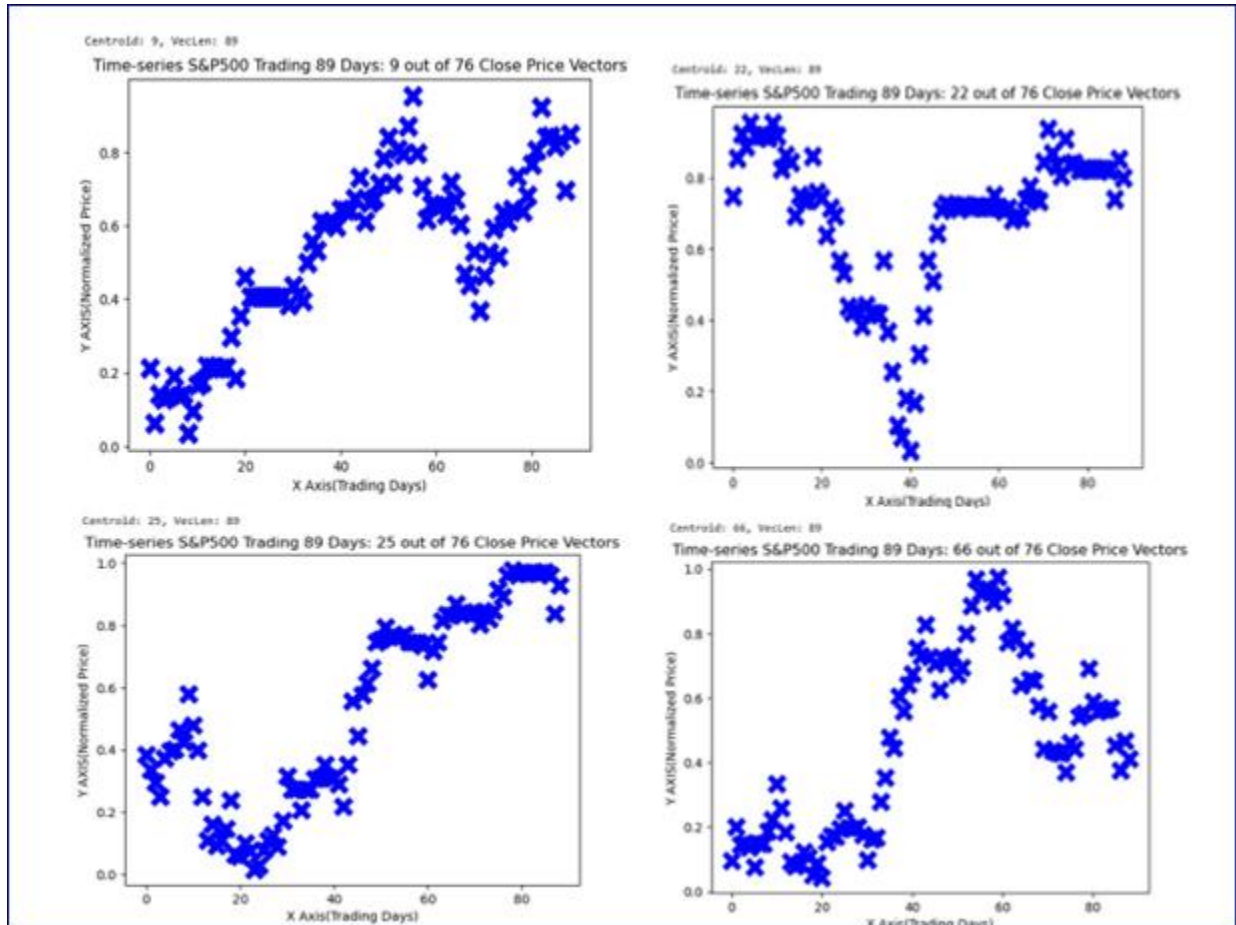


Figure 6: Step 5 Sample charts – 9, 22, 25, 66 Clusters

### Python Code

```
print(' Part 1, Step 9 - Create plots of centroid for each cluster to the screen and to storage')
centroids_x = range(0, VecLen)
for i in range(num_clusters):
    image_path = f"D:/Training_Data/clusterPlots/cluster{i}_plot.png"
    print(image_path)
    # Assuming centroids is a NumPy array with shape (n_clusters, n_features)
    centroids_y = centroids[i]
    plt.scatter(centroids_x, centroids_y, marker="x", s=150, linewidths=5, zorder=10, c=['blue'])
    plt.xlabel("X Axis(Trading Days)")
    plt.ylabel("Y AXIS(Normalized Price)")
    plt.title(f"Time-series S&P500 Trading {VecLen} Days: {i} out of {n_clusters} Closing Price
Vectors")
    # Save the scatter plot as an image
    plt.savefig(image_path)
    plt.show()
```

## Step 6 – Assess Pattern Using GenAI

Step 6 is the longest step because you are: 1) converting the image to base64, 2) configuring the GenAI model, and 3) evaluating the results by the GMT-4o model including recommending the type of pattern and the rationale for its recommendation.

Before we can apply the OpenAI API (Application Program Interface), encode the chart to base64. Base64 is a widely recognized standard for encoding binary data into ASCII characters. This process helps ensure consistent data interpretation across different systems and platforms.

### *Python Code:*

*# Images encoded to base64*

*def encode\_image(image\_path):*

*with open(image\_path, "rb") as image\_file:*

*image\_content = image\_file.read()*

*base64\_string = base64.b64encode(image\_content).decode("utf-8")*

*return base64\_string*

The next task is to provide the parameters required by the OpenAI's API. The API provides the connectivity between the local Python program and the external OpenAI website. The API generates a response based on the parameters provided to the TimeSeriesK-Means model. Parameters required for the successful execution of the GenAI process may include:

- **Model:** Specifies which language model to use (e.g., gpt-4o).
- **Messages:** A list of messages that define the conversation. The messages include the prompt, system role and user roles.
- **Other Parameters:** Such as temperature, max tokens, and top\_p, which controls the generation's behavior, length, and ending conditions.

The output is a generated response that continues the conversation based on the cluster's classification and classification rationale. The user role contains both the prompt and centroid chart.

### *Python Code:*

*for i in range(n\_clusters):*

```
# Encode your image  
image_path = f"D:/Training_Data/clusterPlots/cluster{i}_plot.png"  
base64_image = encode_image(image_path)  
response = openai.chat.completions.create(  
    model = "gpt-4o",  
    messages = [  
        {"role": "system", "content": "The assistance is a Technical Trading Analyst that can  
identify different technical trading patterns."},  
        {"role": "user", "content": prompt},  
        {"role": "user", "content": base64_image}  
    ],  
    max_tokens = max_tokens,  
    temperature = temperature,  
    top_p = top_p  
)  
analysis = str(response.choices[0].message.content.strip())  
print(f"Cluster: {i}, \n Analysis: {analysis} \n\n")
```

The prompt is one of the most important pieces of information in the Messages section. Prompt Engineering is the process of crafting a problem description to obtain the desired results. There are three types of prompts each requiring additional descriptive information. The three types of prompts are: Zero-shot, Few-Shot, and Chain-of-Thought-shot prompting. For this exercise, Zero-shot prompting is adequate to obtain the desired results.

**Python Code:**

```
# Create the prompt with the image
```

```
prompt = """
```

```
I am an associate researcher whose job is to assign names (Classifications) to images. Input is: The  
images are like Technical Trading Analysis pattern charts. The images x-axis in the number of trading  
days from 0 to 89 trading days. The y-axis is the closing price for the trading day, normalized with a  
value between 0 and 1. Process: If the image is like a known technical trading pattern, then assign the  
appropriate technical trading name for the image and the rationale for assigning the classification, if  
the image is not like any technical trading pattern, then assign 'Random' for its image name. Output is:  
The output consists of two fields: image name and its rationale for the classification. If the classification  
is not 'Random' then create a small chart from the same input but highlight the rationale indicators  
with markers to show decision points.  
"""
```

Output from four sample clusters listed below and for the charts in Step 5. The output consists of the Cluster number for cross-referencing the cluster's centroid to the OpenAI GPT-4o generated label. A quick observation from the testing and results indicated: 1) different models produced different classifications, 2) different model settings created different results, 3) Temperature and top\_p not supported for imaging, 4) different results based on available information, 5) the most accurate

classification provides most rationale. Figures 7 through 10 are the results from OpenAI gpt-4 model used to classify the charts Figure 6.

### 1) Results for Chart #9

**Completion 1: \*\*Image Name: \*\* Head and Shoulders**

**\*\*Rationale for Classification: \*\*** The image resembles a Head and Shoulders pattern, which is characterized by three peaks: a higher peak (head) between two lower peaks (shoulders). This pattern suggests a reversal in the trend. In the chart, the left shoulder is formed between days 10 and 20, the head peaks around day 40, and the right shoulder forms between days 60 and 70. The neckline, which connects the lows of the two troughs, can be drawn from around day 15 to day 65, indicating a potential breakdown point.

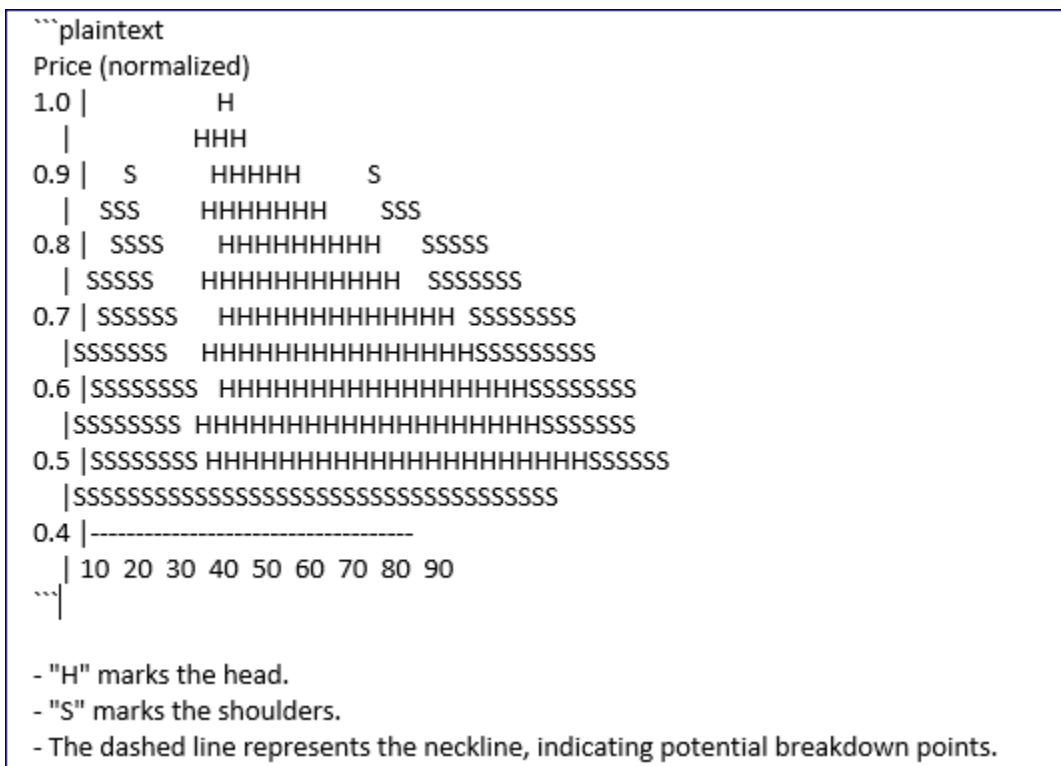


Figure 7 Chart with Rationale Indicators:

### 2) Results for Chart #22

**Completion 1: \*\*Image Name: \*\* Cup and Handle**

**\*\*Rationale for Classification: \*\*** The image resembles the "Cup and Handle" pattern, characterized by a "U" shape followed by a smaller consolidation resembling a handle. The pattern typically indicates a potential bullish continuation. The "cup" forms as the price gradually declines and then rises to the same level as the initial decline, creating a rounded bottom. The "handle" forms as a short-term consolidation or slight pullback after the cup. This pattern is often seen as a precursor to a breakout to the upside. Here is a small chart highlighting the decision points:

1. **Cup Formation:** The initial decline and subsequent rise create a rounded bottom, forming the "cup."
2. **Handle Formation:** A slight pullback or consolidation after the cup, forming the "handle."
3. **Breakout Point:** The point where the price breaks above the resistance level formed by the highs of the cup, indicating a potential upward move.

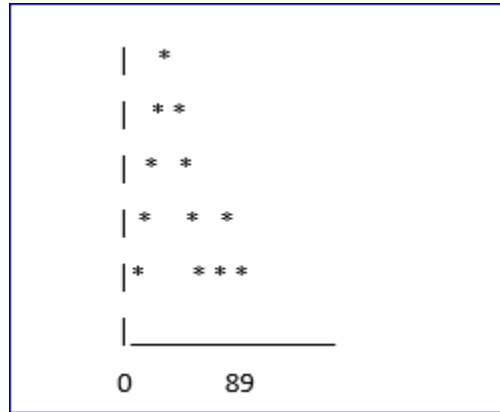


Figure 8 Cup and Handle

- The asterisks (\*) represent the price movement forming the "cup" and "handle."
- The underscore (\_) represents the support and resistance levels.

## #) Results for Chart #25

### Completion 1: **Image Name: Head and Shoulders Pattern**

**Rationale for Classification:** The image resembles a "Head and Shoulders" pattern, which is a common reversal pattern in technical analysis. This pattern is characterized by three peaks: the middle peak (head) being the highest, and the two outside peaks (shoulders) being lower and equal. The pattern is typically associated with a change in trend from bullish to bearish. In the image, the x-axis (trading days) shows a series of peaks where the middle peak is the highest, followed by a decline, which is indicative of this pattern.

**Chart with Rationale Indicators:**

Below is a simplified chart highlighting the decision points for the "Head and Shoulders" pattern:

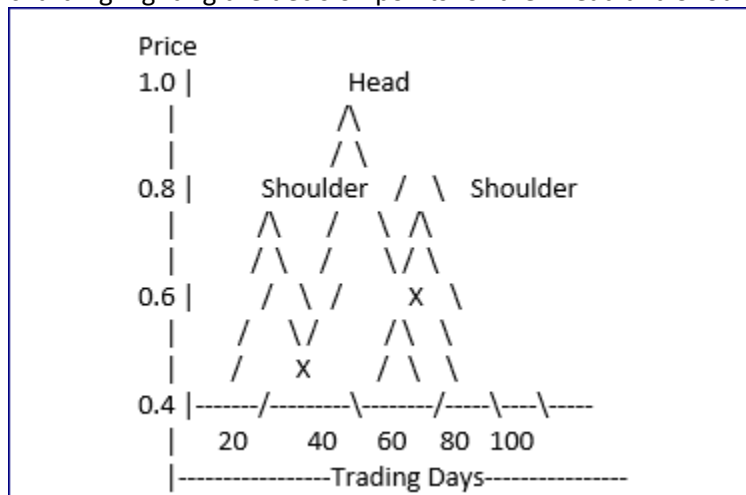


Figure 9 Head and Shoulders Pattern

- The "X" marks the neckline, which is a key level of support that, when broken, confirms the pattern.
- The shoulders are marked at approximately 20 and 80 trading days, and the head is at around sixty trading days.
- The decline after the head and the break below the neckline indicate a potential trend reversal.

=====

#### 4) Results for Chart #66

##### Completion 1: **\*\*Image Name: Head and Shoulders Pattern\*\***

**\*\*Rationale for Classification: \*\***

The image depicts a classic "Head and Shoulders" pattern, which is a popular reversal pattern in technical analysis. This pattern is characterized by three peaks: the middle peak (the "head") being the highest, flanked by two smaller peaks (the "shoulders"). The pattern typically indicates a reversal from a bullish trend to a bearish trend.

**\*\*Highlighted Rationale Indicators: \*\***

1. **\*\*Left Shoulder: \*\*** The first peak is formed, followed by a decline.
2. **\*\*Head: \*\*** The second peak is the highest point in the pattern, followed by another decline.
3. **\*\*Right Shoulder: \*\*** The third peak is formed, which is lower than the head but similar in height to the left shoulder.
4. **\*\*Neckline: \*\*** A support line connecting the lows following the left shoulder and the head. The pattern is confirmed when the price breaks below this neckline after forming the right shoulder.

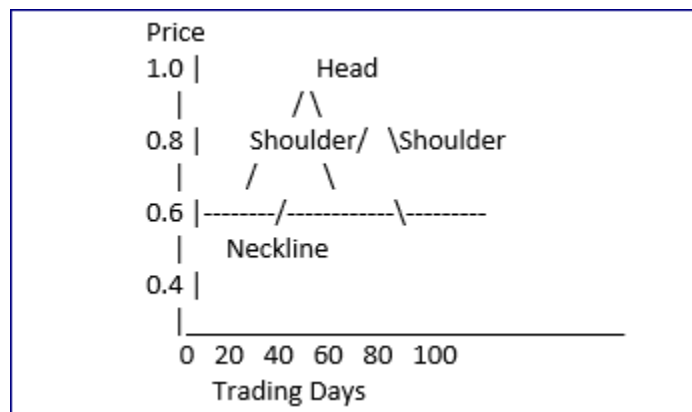


Figure 10: Head, Shoulder, and Neckline Pattern

The chart above illustrates the Head and Shoulders pattern, with markers indicating the left shoulder, head, right shoulder, and neckline.

#### Step 7 – Assign Labels to Patterns

Assigning labels is currently a manual process for validation purposes. This includes ensuring that most if not all possible technical trading patterns are identifiable. The most recent test run of the system identified ten different technical patterns with multiple instances, these patterns include:

1. Head and Sholders

## Emergence of AI Qualitative with Quantitative Data Analysis

2. Inverted Head and Sholders
3. Double Top
4. Triple Top
5. Double Bottom
6. Ascending Triangle
7. Descending Triangle
8. Symmetrical Triangle
9. Cup and Handle
10. Flag

In the above charts, the accuracy varies from excellent to questionable. The desired accuracy is a function of the problem requirements and achieved through tuning. The best to worst models for identifying technical patterns are: OpenAI gpt-3.5 -Turbo, gpt-4, and gpt-4o, and gpt-4o-mini. Further testing will include different models.

See Figure 11 for a partial list of configuration settings for optimization and tuning purposes. Tuning consists of changing the configuration options called hyperparameters and prompt engineering.

Parameters	Options	Settings
Vector Length	89, 55, 34, 21, 13	89
Vector Overlay	21, 13, 8, 5, 3 2	21
Vector Dimension	1, 2, 3	1
Error Function	inertia	inertia
Cluster	K-Means	TimeSeriesK-Means
Data Normalization	Min-Max [0 – 1]	Min-Max [0 – 1]
Model	GPT-3.5, GPT-4	GPT-4o, GPT-4o-mini
Cluster Size	56, 34, 21, 23	56
Token Count	125, 256, 4096	4096
Temperature	Value between 0 & 1	.7 Loose Interpretation
Prompt Type	Zero-shot, Few-shot	Zero-shot
Top_p	Value between 0 & 1	.8

Figure 11 - Critical Configuration Settings

## Concluding Remarks

This paper presented a seven-step method to explore the capabilities of applying GenAI image processing to Technical Trading Analysis providing an increased value proposition for the investor. See

Figure 12 for an overview of the system presented in this paper and the relationship to the follow-up paper: Technical Trading Analysis with Classification, Part II.

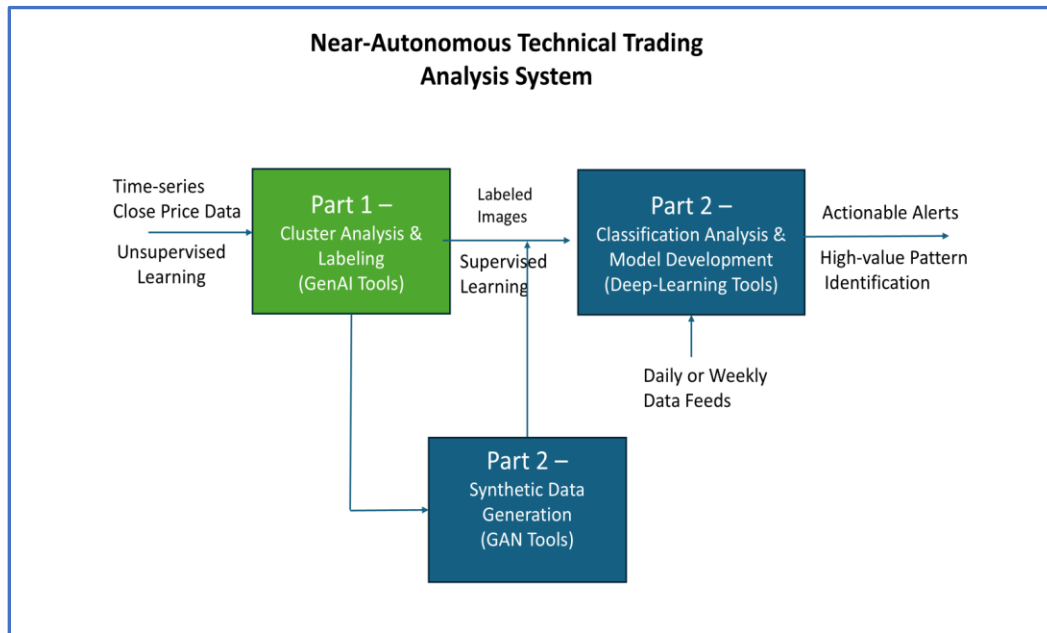


Figure 12 - Summary of Proposed Methodology Highlights

The product of the seven steps is the input to Part II for building Technical Trading Analysis models. The purpose of the model is to identify technical patterns in a timely and accurate manner. The output of Part II will be a Heatmap of analysis results performed at three time periods: Short-, Medium-, and Long-term and for each period covering the market, sector, and equity scenarios.

The objectives and enhancements of Parts I and II are as follows:

**Part I Demonstrated Objectives:**

- Created vectors from historical equity closing prices, over 500 to 1,200 vectors created.
- Used Fibonacci sequence for selecting vector length and overlay length/overlap. combinations evaluated included: 8/3, 13/5, 21/7, 34/9, 89/21.
- Clustered the vectors and calculated the centroid, successfully, continue evaluate centroid trained models in Part II.
- Used the centroid to represent a Technical Trading Chart, successful visualization.
- Identified the chart type, e.g., Head and shoulders, and the rationale for its decision, visual inspection of charts compared to GenAI classification indicate that the OpenAI gpt-4o and gpt-4o-mini successfully classified ten different technical trading charts.

**Part II Planned Enhancements:**

## Emergence of AI Qualitative with Quantitative Data Analysis

- Tune the pre-trained GenAI model and Deep Clustering.
- Use Few-shot prompting for extensible pattern classification capability.
- Create Confusion Matrix analysis by Backtesting.
- Create a Multinomial Logistic Regression model for trading patterns classifications.
- Add 'Trade on the News' functionality – Identifies anomalies/outliers that can disrupt the most correct trading patterns.

Lastly, an opportunity to share lessons learned from this project include:

- Pros:
  - GenAI ability to understand the meaning of words and communicate in conversations.
  - LLMs and pre-trained GenAI models
  - Large inventory of open-source software code and GenAI ability to write code
  - Ability to provide solution through problem description and examples.
- Cons:
  - Lack of backward compatibility for variations in libraries – Moving to a newer version will be costly.
  - Challenge to test, tune, and debug related to backbox-like characteristics. – these tasks are more time consuming.
  - New technical vocabulary - increases the learning curve.
  - Lack of standards by the major players - difficult to move from one GenAI vendor to another.
  - Computer time and modeling resources can be costly – this project OpenAI processing costs over \$300 for this project.

## References

1. OpenAI, Personal Communication, July 7, 2024, [www.openai.com](http://www.openai.com)
2. OpenAI. *ChatGPT* (gpt-4o-2024-08-06) [Large language model], 2024, <https://chat.openai.com>
3. Great Learning, Artificial Intelligence & Machine Learning, May 16, 2020,
4. Great Learning, Generative AI for Natural Language Processing, August 1, 2024
5. Great Learning, ML Foundations for Generative AI, July 13, 2024
6. OpenAI, Quickstart tutorial - OpenAI API, August 4, 2024, [www.openai.com](http://www.openai.com)
7. Slingacademy, Pandas: Create a DataFrame from a NumPy 2-dimensional array, February 19, 2024, [www.slingacademy.com](http://www.slingacademy.com)
8. Oddmund Groette, Technical Analysis Trading Strategy (Rules, Backtest And Example), April 7, 2024 April 7, 2024, <https://www.quantifiedstrategies.com>
9. Arjun Remesh, The Advantages and Disadvantages of Technical Analysis, Updated on March 11, 2024, <https://www.strike.money/>

## Biography

### Mr. Warren V. Camp

Mr. Warren V. Camp is a 30+ subject area expert in Information Systems, Cybersecurity, and Machine Learning. Mr. Camp has obtained college degrees in B.S. in Mathematics, M.S. in Computer Science, and MBA in Finance, certifications in CPA, CISSP, CISA, CISM, CEH, and online courses in machine learning, AWS Bedrock, AZURE Machine Learning, and GenAI. Over 15 years teaching as an adjunct professor at Tulane, National Louis, and University Maryland Global Campus universities teaching courses in Information Technology, Cybersecurity, and Finance.

His first AI project was developing a 4x4x4 Tic-Tac-Toe game, with 64! possible moves required complex decision tree strategy, published article on Small Word Size Random Number Generator, developed Cross-Compiler of Complex Instruction Set Computer architecture to a Stack-based CPU Organization, developed stock market prediction application using MATLAB, worked several year doing data analysis and machine learning models for detecting violations of NASDAQ and New York Stock Market compliance regulations.

### Mr. Lopez-Gomez

Mr. Lopez-Gomez is a 30+ years as seasoned technology executive with a distinguished career as a Chief Technology Officer (CTO) and Chief Information Officer (CIO) of an independent federal agency. With deep expertise in cybersecurity, he has led and safeguarded critical IT infrastructures across various sectors, including a notable tenure as a DOJ federal employee and consultant with DHS and DoD. Mr. Lopez-Gomez continues to share his wealth of knowledge and experience, ensuring that organizations remain secure and innovative in an ever-evolving digital landscape.