# Feasibility for Utilizing Engineered Financial Network features for Predicting Markets' Next Day Price Movements

Gustavo Zambrana

June 2020

# Agenda

- Abstract
- Project Objectives
- Validation Methodology
- Theoretical Summary
- Empirical Results
- Conclusions
- Recommendations

# Abstract

- Stock Market predictions is one of the most research field in Finance. The main objective of such studies is predicting the price directions of specific stocks or market indexes.

- Per Reference 1, most of the research and corresponding analysis involves utilizing different measures for returns and volatility, namely, daily, 5-day moving average returns, etc.

- The authors of Reference 1 claimed that there is seldom consideration for co-movement among stocks in the same market, and consequently a topic of research that deserves evaluation.
  - The analysis conducted cannot ascertain the validity of such claim.

# Project Objectives

- Access the validity of Reference’s 1 claim:
  - Engineered Network Topology Characteristics for a set of related markets' indexes can be successfully utilized in forecasting the direction of price movements using Machine Learning algorithms
  - The authors of Reference 1 evaluated a set of related markets, namely, S&P 500, Dow Jones Industrial, and Nasdaq Composite as components of a complex network.
- Evaluate validity of the approach with other markets and time periods
- Analyze the impact of the Engineered Network derived features on forecasting
- Evaluate additional Machine Learning algorithms not studied by Reference 1
- Analyze the distribution of price movements of the studied markets to ensure that forecasting results provide ”learning” value

# Validation Methodology

1. **Validate results stated by Reference 1 utilizing the same market indexes, time periods, and daily data, as well as the same exact network engineered measures and derivations employed**
    - Validate derived Network Topology features by comparison to graphs of the same parameters presented in Ref. 1
    - Application of the same Machine Learning algorithms (KNN, SVC) , cross validation with the same size, and Training and Test data sets

2. **Execute additional experiments to validate or negate the results, including:**
    - Conduct the same experiment as validation step 1 against a complete different set of market indexes, and different time periods for the training and test sets
    - Evaluate the importance or lack-thereof of the derived Network Topology features on forecasting
    - Explore the use of additional Machine Learning methods no evaluated by Reference 1

3. **Determine additional classification metrics no evaluated by Reference 1**
    - Precision
    - Recall

4. **Analyze the distribution of price movements for the markets' studied**
    - Plot indexes classifications versus time, as well as histograms of each index classifications
    - Compute transition statistics for all indexes from one classification to another (16 possible transitions)
    - Compute aggregate transition statistics for all indexes.
        - S1S2 → S1S2, S1S2 → S3S4, S3S4 → S1S2, S3S4 → S3S4
    - Pareto Charts to visualize
    - Analysis of Daily Log Returns per Index Classification transitions for each index

# Theoretical Summary

- Formulas for derived Financial features

$$r = \ln \frac{Close\,(t)}{Close\,(t-1)},$$

$$R = \ln \frac{Close\,(t)}{Close\,(t-5)},$$

$$\mathrm{V} = std\,(r_1, \ldots, r_5) * \sqrt{5},$$

$$\mathrm{V}' = \frac{1}{M}\sum_{i=1}^{M}\overline{V}_i.$$

Where:

- Close(t) = Closing Price
- r = daily log return
- R = five-day log return
- V = five-day volatility
- V’ = Average five-day volatility of the markets analyze

- Stocks’ Classifications

$$f = \begin{cases} S1, & if\ R \geq 0\ and\ V \geq \mathrm{V}' \ \text{(sharp rise)} \\ S2, & if\ R \geq 0\ and\ V < \mathrm{V}' \ \text{(stable rise)} \\ S3, & if\ R < 0\ and\ V < \mathrm{V}' \ \text{(stable decline)} \\ S4, & if\ R < 0\ and\ V \geq \mathrm{V}' \ \text{(sharp decline)}. \end{cases}$$

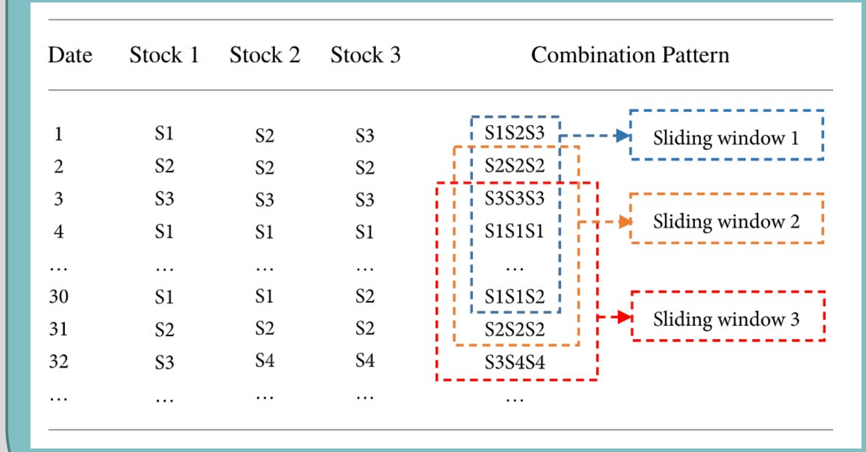
# Theoretical Summary

➢ Derivation of Engineered Network Features per Day

1. Create Node per Day by combining each stock's classification
2. Create a Network with a Sliding Window of 30 days
3. Compute Network Measures for each sliding window
4. Assign engineered Network measures of the sliding window to the end of the window + 2 days (to prevent leakage)

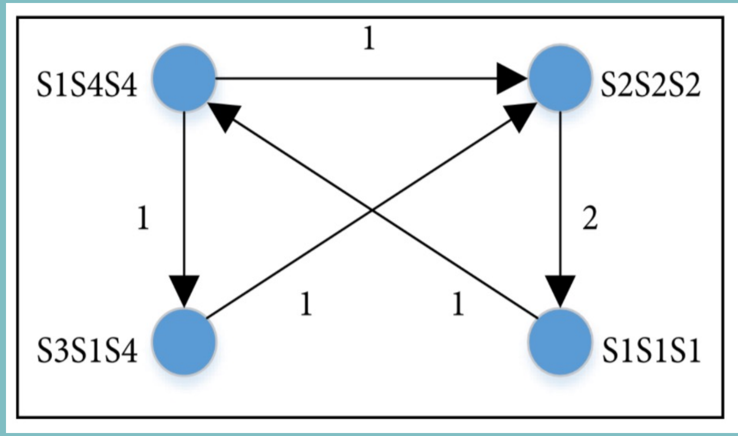
## Process to create Engineered Network Measures per Day

Create Sliding Windows → Create Network per Sliding Window → Compute Network Measures



| Date | Stock 1 | Stock 2 | Stock 3 | Combination Pattern |
|---|---|---|---|---|
| 1 | S1 | S2 | S3 | S1S2S3 |
| 2 | S2 | S2 | S2 | S2S2S2 |
| 3 | S3 | S3 | S3 | S3S3S3 |
| 4 | S1 | S1 | S1 | S1S1S1 |
| ... | ... | ... | ... | ... |
| 30 | S1 | S1 | S2 | S1S1S2 |
| 31 | S2 | S2 | S2 | S2S2S2 |
| 32 | S3 | S4 | S4 | S3S4S4 |
| ... | ... | ... | ... | ... |






Network Average Degree Centrality
Average Network Strength
Network Average Shortest Path Length
Network Closeness Centrality

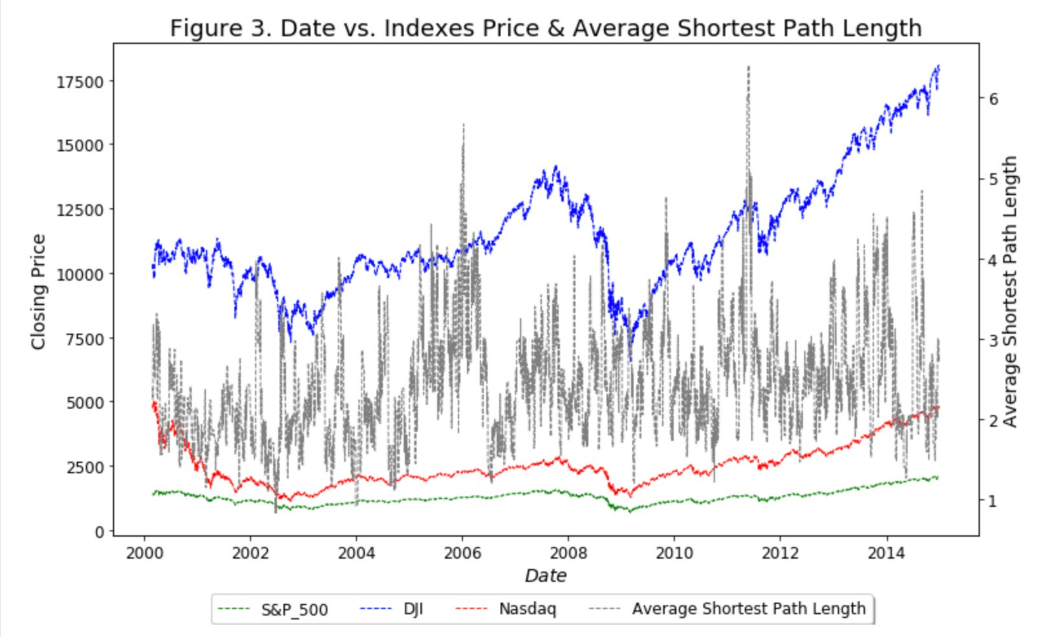
# Empirical Results

- Validation of Derived Engineered Network Measures
- Analysis of Exploratory Experiments
- Optimized Machine Learning Training Sets' Results
- Optimized Machine Learning Testing Sets' Results
- Analysis of Price Movements for Analyzed Markets with respect to Indexes' Classification
- Cluster Analysis with Unsupervised Learning for Stock Index Classification

# Empirical Results : *Validation of Derived Engineered Network Measures*

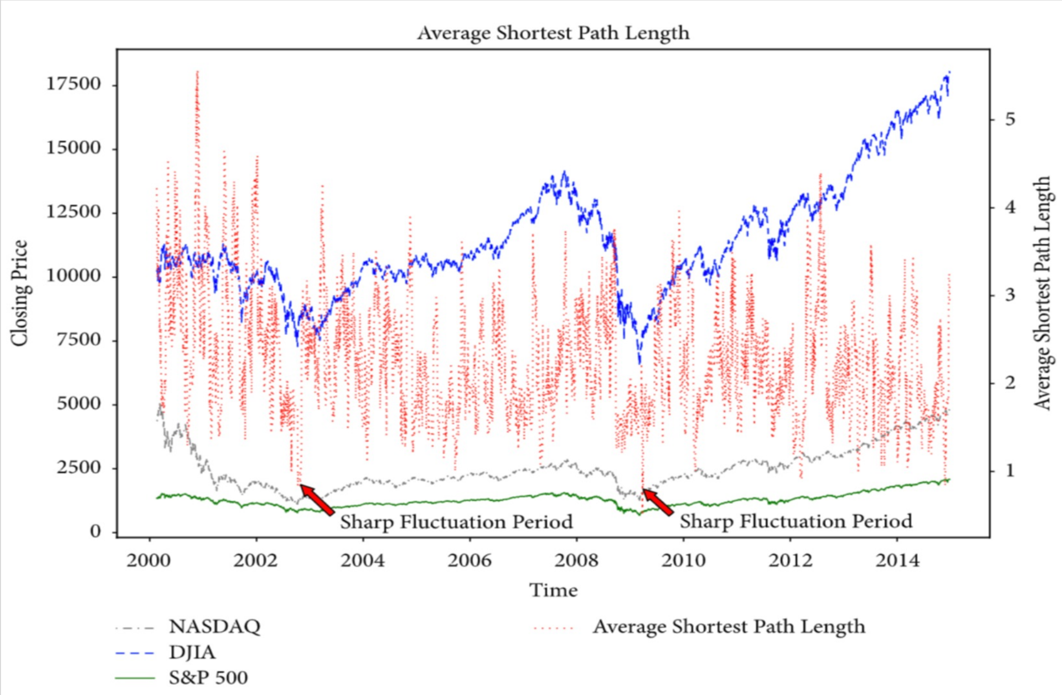
### Derived *Average Shortest Path Length*

(Fig 3 Jupyter Notebook)




### Ref. 1 *Average Shortest Path Length*

(Fig 6)




For additional comparison and analysis of all Engineered Network Measures, see 6.2 Analysis of Network Topological Characteristics

# Empirical Results : *Analysis of Exploratory Experiments*

- Performance comparison between two distinct Datasets

| Dataset | Indexes | Training Set Dates | Testing Set Dates |
|---|---|---|---|
| #1 | DJI, S&P 500, Nasdaq | Jan 2000 to Dec 2014 | Jan 2015 to Dec 2017 |
| #2 | MXUS, MSDUUK, MSDUJN | Jan 2002 to Dec 2012 | Jan 2013 to Jul 2015 |

- *Performance comparison Between different feature sets*

| Feature Set | Description |
|---|---|
| #1 | Evaluate Only Financial derived Index Features |
| #2 | Evaluate Only Network Features |
| #3 | Evaluate all Network Features & all Financial Index Measures |

- *Performance comparison Between Classifiers*
    - *KNN, SGD, SVC, Random Forest*

Refer to section 6.3.4.2 Analyze Experiment's Results

# Empirical Results : *Analysis of Exploratory Experiments*

- Performance comparison between the two distinct Datasets (~1500 simulations per dataset)

**Dataset 1: Mean Values of Precision, Recall, and Accuracy for Combine ML Model**

(Results are group by: Experiment Type and Classifier)

```
dat1_results_stats.style.format("{:.2%}")
```

| Experiment Description | Classifier | Combine_ML_Precision | Combine_ML_Recall | Combine_ML_Accuracy |
|---|---|---|---|---|
| Evaluate Only Financial Index Features | KNeighborsClassifier | 76.37% | 67.36% | 76.02% |
| | RandomForestClassifier | 75.49% | 69.84% | 76.30% |
| | SGDClassifier | 78.28% | 71.22% | 78.22% |
| | SVC | 78.62% | 73.32% | 79.11% |
| Evaluate Only Network Features | KNeighborsClassifier | 41.42% | 33.89% | 48.83% |
| | RandomForestClassifier | 40.10% | 36.18% | 47.05% |
| | SGDClassifier | 41.96% | 29.66% | 50.08% |
| | SVC | 22.46% | 2.02% | 52.84% |
| Evaluate all Network Features & all Financial Index Measures | KNeighborsClassifier | 73.11% | 62.67% | 72.90% |
| | RandomForestClassifier | 74.28% | 67.74% | 75.01% |
| | SGDClassifier | 74.93% | 69.61% | 75.88% |
| | SVC | 76.79% | 71.11% | 77.40% |

# Empirical Results : *Analysis of Exploratory Experiments*

- Performance Comparison Between different feature sets

**Mean Values of Precision, Recall, and Accuracy for Combine ML Model all datasets**

(Results are group by: Experiment Type)

```
results_stats.style.format("{:.2%}")
```

| Experiment Description | Combine_ML_Precision | Combine_ML_Recall | Combine_ML_Accuracy |
|---|---|---|---|
| **Evaluate Only Financial Index Features** | 77.04% | 70.22% | 76.91% |
| **Evaluate Only Network Features** | 41.01% | 28.70% | 50.73% |
| **Evaluate all Network Features & all Financial Index Measures** | 75.30% | 68.33% | 75.40% |

# Empirical Results : *Analysis of Exploratory Experiments*

## Summary of Experiments' Results

- There are only minor differences due to difference datasets, that is, different market indexes, and different time periods
- The best label to use for predicting price movement patterns, it is the Multi Label, combining S1 and S2, and S3 and S4 stock classifications using the confusion matrix
  - S1S2 indicate next day return is positive
  - S3S4 indicate next day return is negative
- The best experiment type is the one with only Financial Indexes derived features
- The best performing models across all evaluated classifiers are SGD and SVC
- The performance of KNN and Random Forest classifiers are similar, but slightly lower than the SGD and SVC algorithms
- Based on the above results, the following Machine Learning algorithms were selected for optimization:
  - KNN
  - Random Forest
  - SGD
  - SVC

# Empirical Results : *Training Sets Results*

- Summary of Results For Optimized Classifiers against the Training Sets Using Cross Validation

| Classifier | Metric | DJI | S&P_500 | Nasdaq | MXUS | MSDUUK | MSDUJN |
|---|---|---|---|---|---|---|---|
| KNeighborsClassifier | Accuracy | 80.74% | 80.55% | 81.60% | 80.95% | 80.67% | 78.41% |
| | Precision | 79.90% | 80.85% | 81.80% | 82.01% | 80.15% | 80.13% |
| | Recall | 75.08% | 74.61% | 76.55% | 74.22% | 76.55% | 72.65% |
| RandomForestClassifier | Accuracy | 81.44% | 80.85% | 81.54% | 81.66% | 81.59% | 79.12% |
| | Precision | 80.04% | 80.29% | 79.47% | 81.31% | 78.77% | 80.35% |
| | Recall | 76.97% | 76.33% | 80.08% | 77.26% | 81.58% | 74.28% |
| SVC | Accuracy | 79.67% | 79.77% | 80.36% | 80.07% | 80.32% | 77.92% |
| | Precision | 80.46% | 79.78% | 81.50% | 79.35% | 81.97% | 79.80% |
| | Recall | 70.99% | 73.96% | 73.48% | 75.70% | 72.83% | 71.76% |
| SGDClassifier | Accuracy | 79.72% | 79.94% | 80.26% | 78.13% | 80.21% | 77.46% |
| | Precision | 81.08% | 81.68% | 81.21% | 83.95% | 79.93% | 80.19% |
| | Recall | 70.26% | 71.65% | 73.60% | 63.94% | 75.54% | 69.90% |

*Refer to section 6.3.6 Summary of Training Set Results*

# Empirical Results : *Testing Sets' Results*

- Summary of Testing Set results for each classifier across all indexes
- Reference 1 Testing Set results
- Comparison of Testing and Training Sets
- Testing set Confusion Matrixes for Best Classifier across Stock Indexes

*Refer to sections 6.3.7 Evaluate Performance with Test Datasets*

# Empirical Results : *Testing Sets Results*

➢ Summary of Results For Optimized Classifiers against the Testing Sets

| Classifier | Metric | DJI | S&P_500 | Nasdaq | MXUS | MSDUUK | MSDUJN |
|---|---|---|---|---|---|---|---|
| KNeighborsClassifier | Accuracy | 76.42% | 75.10% | 78.68% | 78.02% | 79.87% | 76.01% |
| | Precision | 69.85% | 68.00% | 71.90% | 67.82% | 78.21% | 71.71% |
| | Recall | 73.94% | 68.92% | 74.58% | 79.02% | 75.85% | 72.55% |
| RandomForestClassifier | Accuracy | 77.88% | 75.23% | 78.54% | 77.35% | 80.37% | 76.01% |
| | Precision | 72.29% | 67.30% | 69.50% | 67.18% | 75.87% | 71.05% |
| | Recall | 73.94% | 71.62% | 80.34% | 77.68% | 81.89% | 74.12% |
| SVC | Accuracy | 77.22% | 74.97% | 78.81% | 79.19% | 81.71% | 76.17% |
| | Precision | 71.99% | 67.77% | 71.70% | 70.49% | 81.45% | 72.51% |
| | Recall | 71.99% | 68.92% | 75.59% | 76.79% | 76.23% | 71.37% |
| SGDClassifier | Accuracy | 78.94% | 75.63% | 79.47% | 80.54% | 82.05% | 75.67% |
| | Precision | 75.52% | 71.05% | 75.55% | 75.71% | 83.19% | 72.54% |
| | Recall | 71.34% | 63.85% | 70.17% | 70.98% | 74.72% | 69.41% |

*Refer to section 6.3.7.2 Summary of Results For Optimized Classifiers against the Testing Sets*

# Empirical Results : *Testing Sets Results*

- Reference 1 Testing Set results

| Algorithms | DJIA stock index | S&P500 stock index | NASDAQ stock index |
|---|---|---|---|
| KNN | 74.83% | 72.58% | 72.45% |
| SVM | 74.97% | 73.11% | 74.57% |

*Refer to Reference 1, Table 3*

# Empirical Results : *Testing Sets Results*

➢ Comparison of Testing and Training Sets (Testing – Training sets differences)

| Classifier | Metric | DJI | S&P_500 | Nasdaq | MXUS | MSDUUK | MSDUJN |
|---|---|---|---|---|---|---|---|
| KNeighborsClassifier | Accuracy | -4.32% | -5.45% | -2.92% | -2.93% | -0.80% | -2.41% |
| | Precision | -10.05% | -12.85% | -9.91% | -14.20% | -1.94% | -8.43% |
| | Recall | -1.13% | -5.70% | -1.97% | 4.80% | -0.70% | -0.10% |
| RandomForestClassifier | Accuracy | -3.56% | -5.61% | -3.00% | -4.31% | -1.22% | -3.11% |
| | Precision | -7.75% | -12.99% | -9.97% | -14.13% | -2.90% | -9.30% |
| | Recall | -3.02% | -4.71% | 0.26% | 0.42% | 0.31% | -0.16% |
| SVC | Accuracy | -2.45% | -4.81% | -1.56% | -0.88% | 1.39% | -1.74% |
| | Precision | -8.47% | -12.01% | -9.80% | -8.86% | -0.52% | -7.29% |
| | Recall | 0.99% | -5.04% | 2.11% | 1.08% | 3.39% | -0.38% |
| SGDClassifier | Accuracy | -0.78% | -4.31% | -0.79% | 2.41% | 1.84% | -1.79% |
| | Precision | -5.57% | -10.62% | -5.66% | -8.23% | 3.26% | -7.65% |
| | Recall | 1.07% | -7.80% | -3.43% | 7.04% | -0.82% | -0.49% |

- **There is overfit across most Classifiers except for SGD (MXUS, MSDUUK) and SVC (MSDUUK)**
- **The range of overfit in percentages is between -0.80% to -5.61% in terms of accuracy**

*Refer to sections 6.3.7.3 Comparison of Testing and Training Sets*

# Empirical Results : *Testing Sets Results*

- Testing set Confusion Matrixes for Best Classifier across Stock Indexes

1. Confusion Matrix for Classifier : KNeighborsClassifier for stock index : MSDUUK

4x4 C.M. For Multi Label (S1,S2,S3,S4)

| | S1 | S2 | S3 | S4 |
|---|---|---|---|---|
| **S1** | 71 | 8 | 6 | 13 |
| **S2** | 135 | 61 | 16 | 21 |
| **S3** | 28 | 17 | 62 | 69 |
| **S4** | 17 | 2 | 15 | 55 |

2x2 C.M. For Merge Multi Label (S1S2, S3S4)

| | S1S2 | S3S4 |
|---|---|---|
| **S1S2** | 275 | 56 |
| **S3S4** | 64 | 201 |

2. Confusion Matrix for Classifier : RandomForestClassifier for stock index : MSDUUK

4x4 C.M. For Multi Label (S1,S2,S3,S4)

| | S1 | S2 | S3 | S4 |
|---|---|---|---|---|
| **S1** | 55 | 15 | 6 | 22 |
| **S2** | 89 | 103 | 26 | 15 |
| **S3** | 16 | 16 | 74 | 70 |
| **S4** | 11 | 5 | 19 | 54 |

2x2 C.M. For Merge Multi Label (S1S2, S3S4)

| | S1S2 | S3S4 |
|---|---|---|
| **S1S2** | 262 | 69 |
| **S3S4** | 48 | 217 |

*Refer to sections 6.3.7.4 Testing set Confusion Matrixes for Best Classifier across Stock Indexes modeled*

# Empirical Results : *Testing Sets Results*

- Testing set Confusion Matrixes for Best Classifier across Stock Indexes

3. Confusion Matrix for Classifier : SVC for stock index : MSDUUK

4x4 C.M. For Multi Label (S1,S2,S3,S4)

| | S1 | S2 | S3 | S4 |
|---|---|---|---|---|
| **S1** | 55 | 26 | 4 | 13 |
| **S2** | 40 | 164 | 18 | 11 |
| **S3** | 14 | 29 | 78 | 55 |
| **S4** | 9 | 11 | 26 | 43 |

2x2 C.M. For Merge Multi Label (S1S2, S3S4)

| | S1S2 | S3S4 |
|---|---|---|
| **S1S2** | 285 | 46 |
| **S3S4** | 63 | 202 |

4. Confusion Matrix for Classifier : SGD for stock index : MSDUUK

4x4 C.M. For Multi Label (S1,S2,S3,S4)

| | S1 | S2 | S3 | S4 |
|---|---|---|---|---|
| **S1** | 80 | 1 | 4 | 13 |
| **S2** | 190 | 20 | 11 | 12 |
| **S3** | 36 | 6 | 74 | 60 |
| **S4** | 24 | 1 | 22 | 42 |

2x2 C.M. For Merge Multi Label (S1S2, S3S4)

| | S1S2 | S3S4 |
|---|---|---|
| **S1S2** | 291 | 40 |
| **S3S4** | 67 | 198 |

*Refer to sections 6.3.7.4 Testing set Confusion Matrixes for Best Classifier across Stock Indexes modeled*

# Empirical Results :

*Analysis of Price Movements for Analyzed Markets with respect to Indexes' Classification*

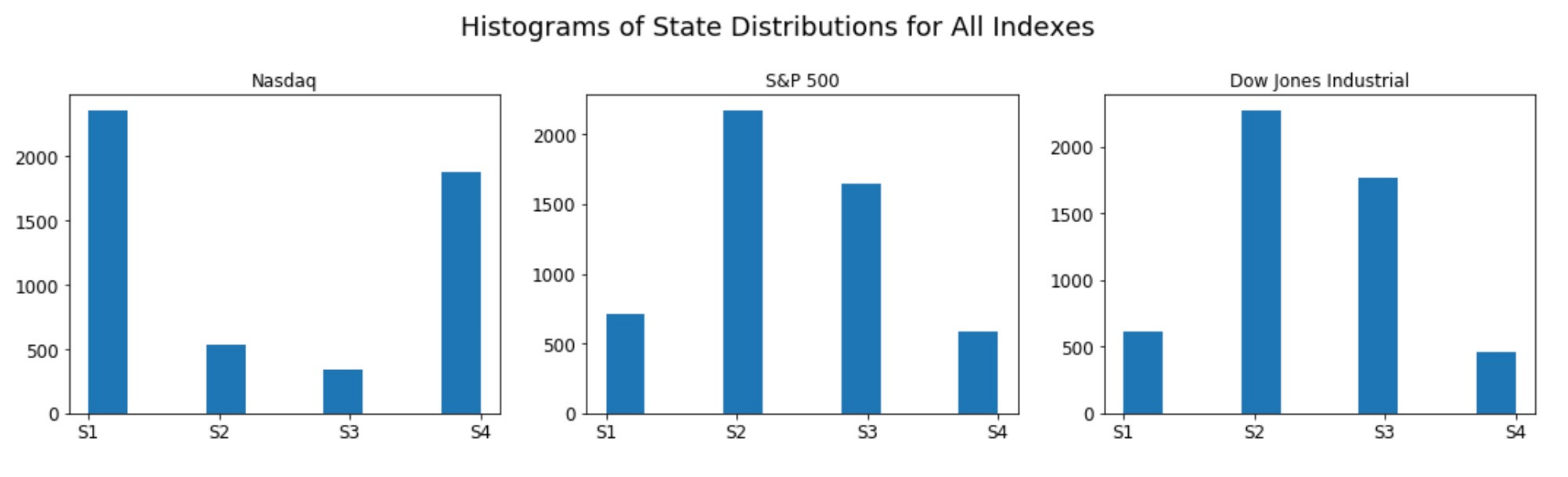
- Histogram of Indexes' Classifications
- Indexes' Classification vs Time Graphs
- *Summary Statistics for Transitions from one Classification to Another Classification*
- *Pareto Charts for Transitions*
- *Analysis of Daily Log Returns per Index Classification*

*Refer to sections 6.1.6 Analysis & Visualization of Stock Indexes Classifications' Distributions*

# Empirical Results :

*Analysis of Price Movements for Analyzed Markets with respect to Indexes' Classification*

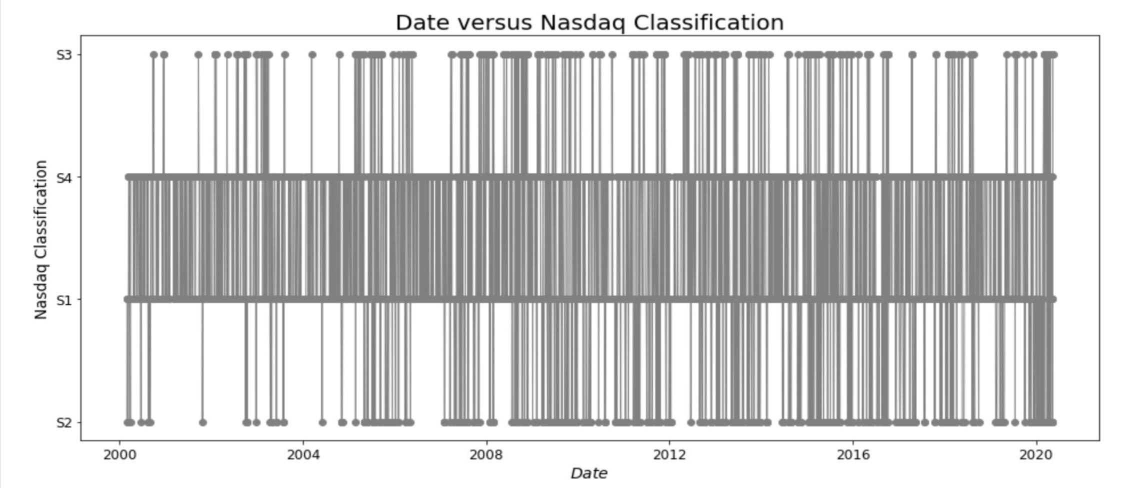
- Histogram of Indexes' Classifications

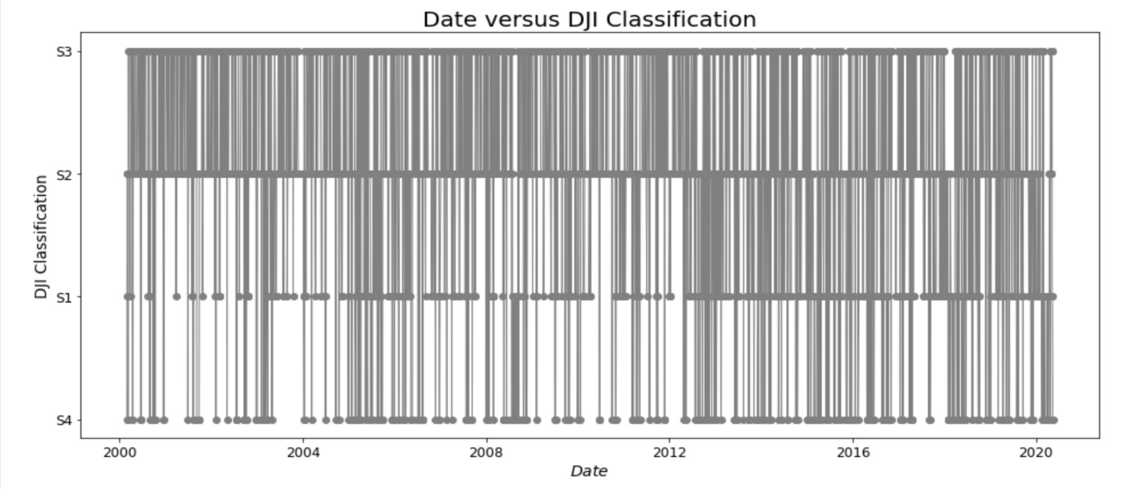


# Empirical Results :

*Analysis of Price Movements for Analyzed Markets with respect to Indexes' Classification*

- Index Classification vs Time ( Nasdaq & DJI)



# Empirical Results :

*Analysis of Price Movements for Analyzed Markets with respect to Indexes' Classification*

- Summary Statistics for Transitions from one Classification to Another Classification

**Detail Transition Classification Summary**

| | Transition | DJI | S&P_500 | Nasdaq |
|---|---|---|---|---|
| 2 | S1_to_S1% | 7.08104 | 8.30055 | 34.6971 |
| 3 | S1_to_S2% | 2.9701 | 2.89142 | 2.93076 |
| 4 | S1_to_S3% | 2.04563 | 2.85208 | 8.37923 |
| 5 | S1_to_S4% | 0 | 0 | 0 |
| 6 | S2_to_S1% | 3.08812 | 2.87175 | 2.93076 |
| 7 | S2_to_S2% | 33.5956 | 32.2581 | 6.03855 |
| 8 | S2_to_S3% | 7.23839 | 6.53029 | 0.767113 |
| 9 | S2_to_S4% | 0.767113 | 0.86546 | 0.688434 |
| 10 | S3_to_S1% | 0.668765 | 0.767113 | 0.629426 |
| 11 | S3_to_S2% | 7.23839 | 6.3336 | 0.826121 |
| 12 | S3_to_S3% | 24.3509 | 22.8954 | 3.54052 |
| 13 | S3_to_S4% | 2.14398 | 1.88828 | 1.77026 |
| 14 | S4_to_S1% | 1.25885 | 2.10464 | 7.7498 |
| 15 | S4_to_S2% | 0.88513 | 1.02282 | 0.609756 |
| 16 | S4_to_S3% | 2.0653 | 1.57356 | 1.67191 |
| 17 | S4_to_S4% | 4.58301 | 6.82533 | 26.7506 |

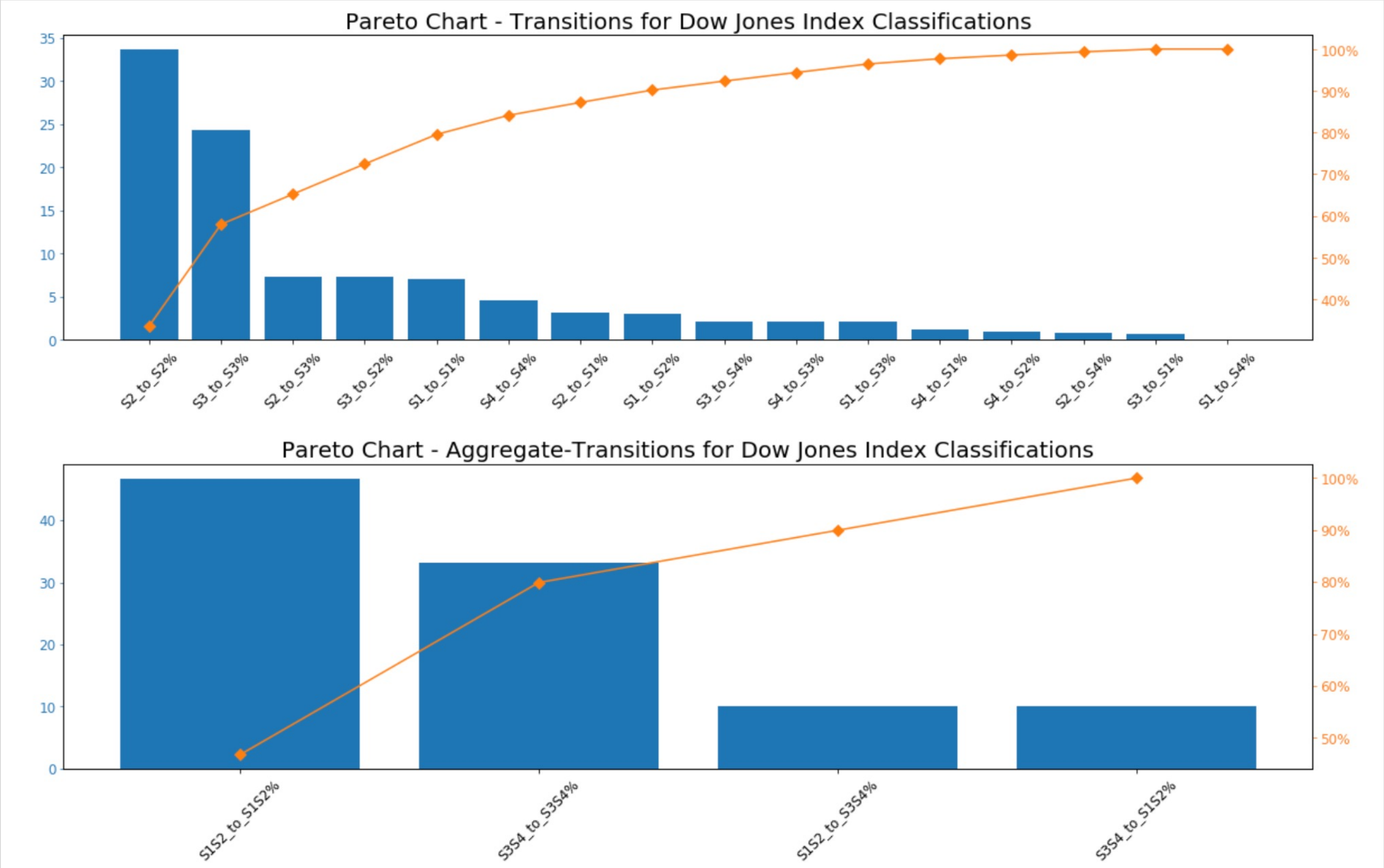
**Aggregate Transition Classification Summary**

| | Transition | DJI | S&P_500 | Nasdaq |
|---|---|---|---|---|
| 2 | S1S2_to_S1S2% | 46.7349 | 46.3218 | 46.5972 |
| 3 | S1S2_to_S3S4% | 10.0511 | 10.2478 | 9.83478 |
| 4 | S3S4_to_S3S4% | 33.1432 | 33.1825 | 33.7333 |
| 5 | S3S4_to_S1S2% | 10.0511 | 10.2282 | 9.81511 |

# Empirical Results :

*Analysis of Price Movements for Analyzed Markets with respect to Indexes' Classification*

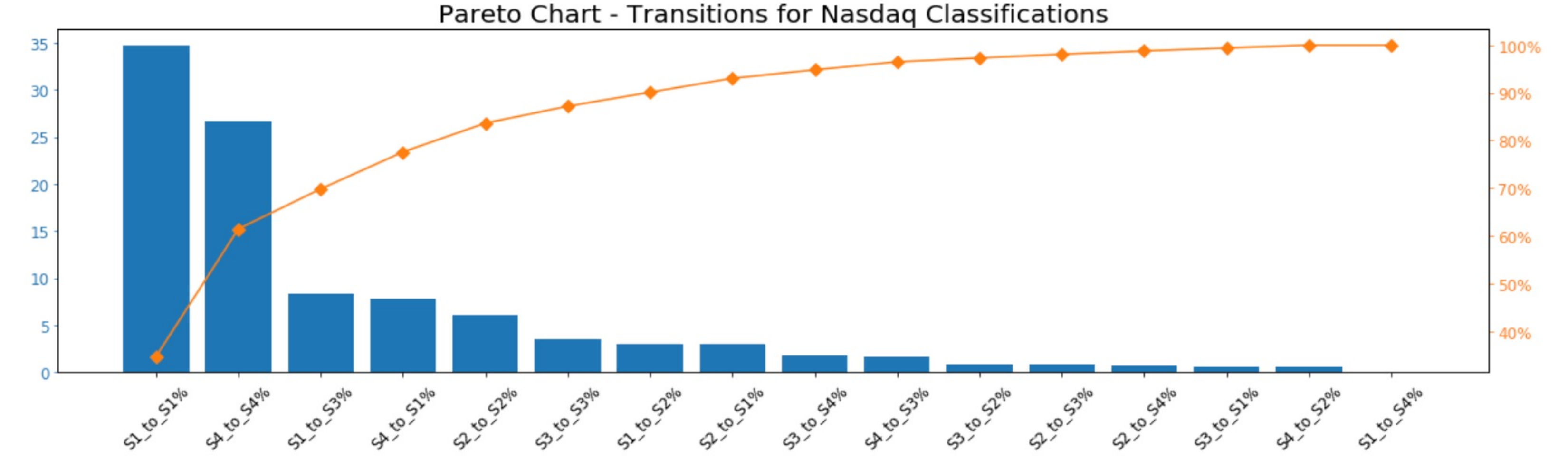
➢ Pareto Charts for Transitions (DJI)

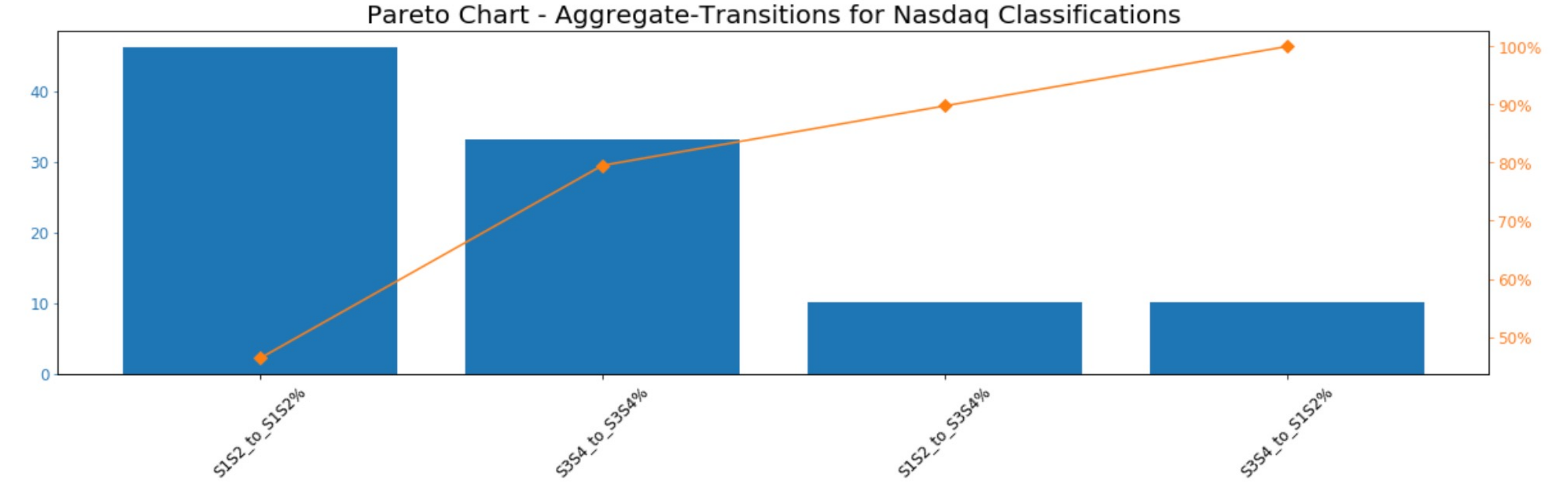


# Empirical Results :

*Analysis of Price Movements for Analyzed Markets with respect to Indexes' Classification*

- Pareto Charts for Transitions (Nasdaq)

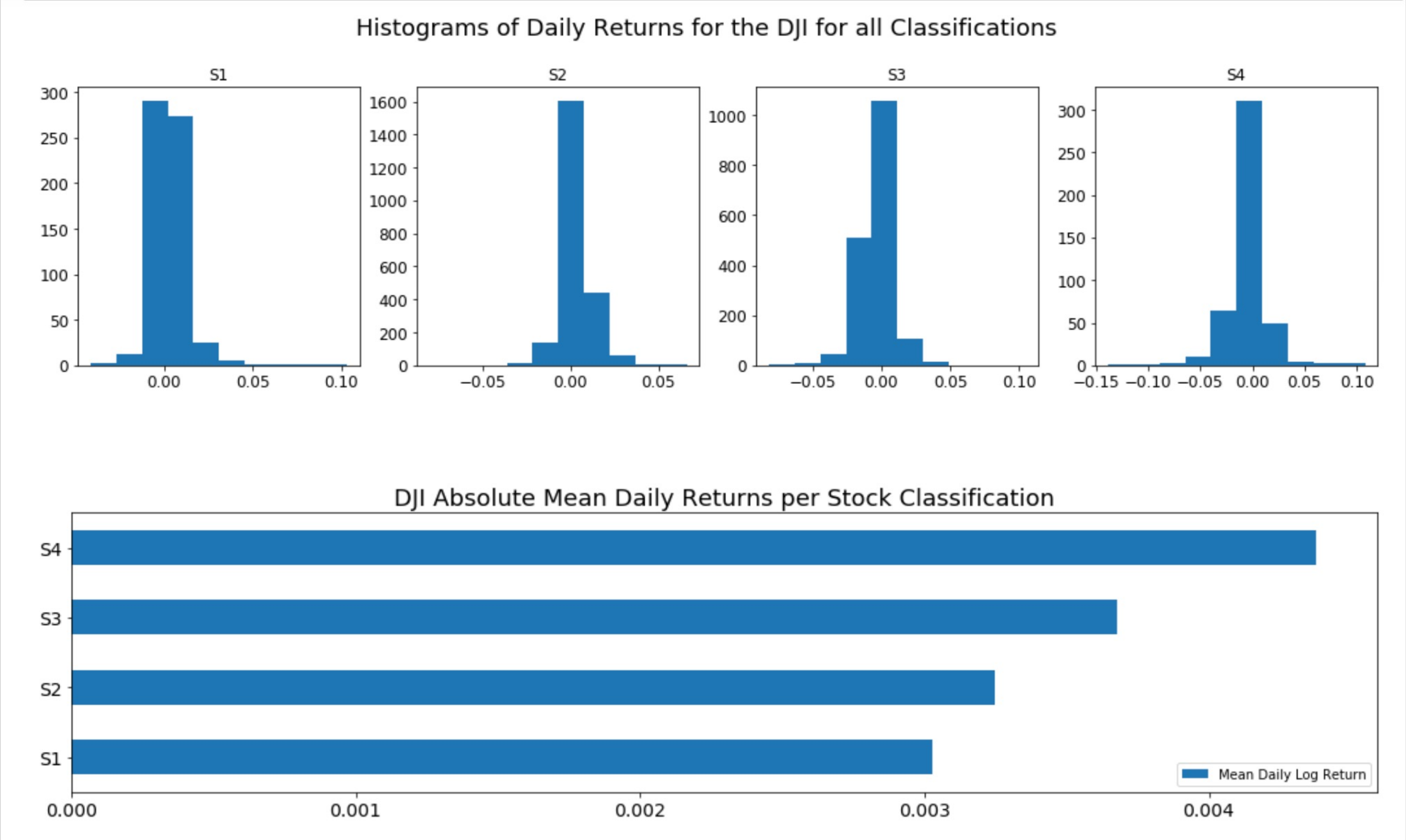


# Empirical Results :

*Analysis of Price Movements for Analyzed Markets with respect to Indexes' Classification*

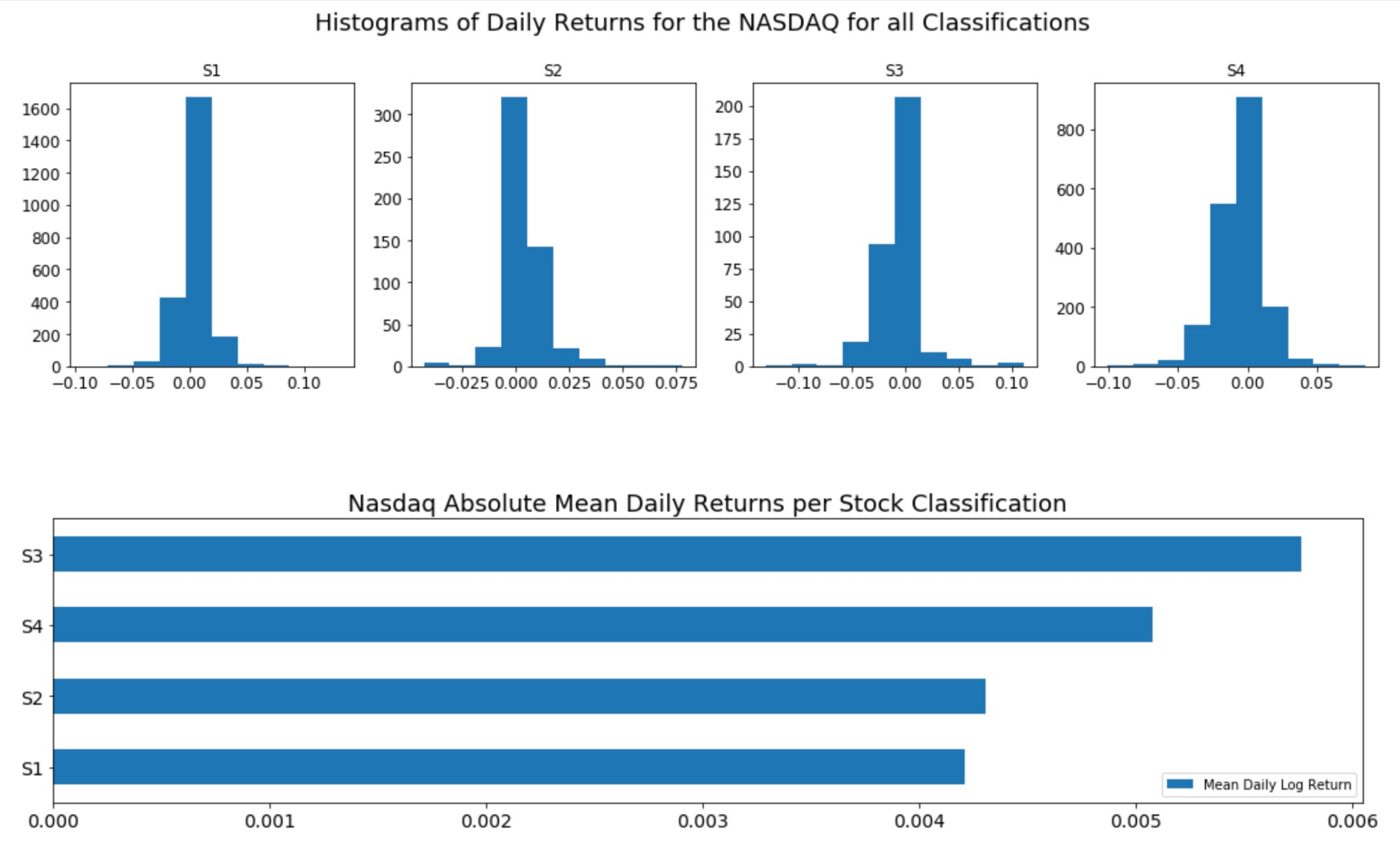
- Dow Jones Industrial



# Empirical Results :

*Analysis of Price Movements for Analyzed Markets with respect to Indexes' Classification*

- Nasdaq



# Empirical Results : *Cluster Analysis with Unsupervised Learning for Stock Index Classification*

- Create Clusters Per Stock Index
- Plot Inertias vs K For All Indexes
- Centroids for all Indexes for K=4

*Refer to sections 6.1.8 Cluster Analysis with Unsupervised Learning*

# Empirical Results : *Cluster Analysis with Unsupervised Learning for Stock Index Classification*

## ➢ Creation of Clusters Per Index

- The analysis consists of deducing the classification clusters for each stock index using the same financial features utilized to create the engineered classification (S1, S2, S3, S4)

- The features utilized are:
  - R → Five-day average daily log return
  - V → Five-day volatility of daily log returns
  - V' → Average volatility across all indexes in market's analyze as a group

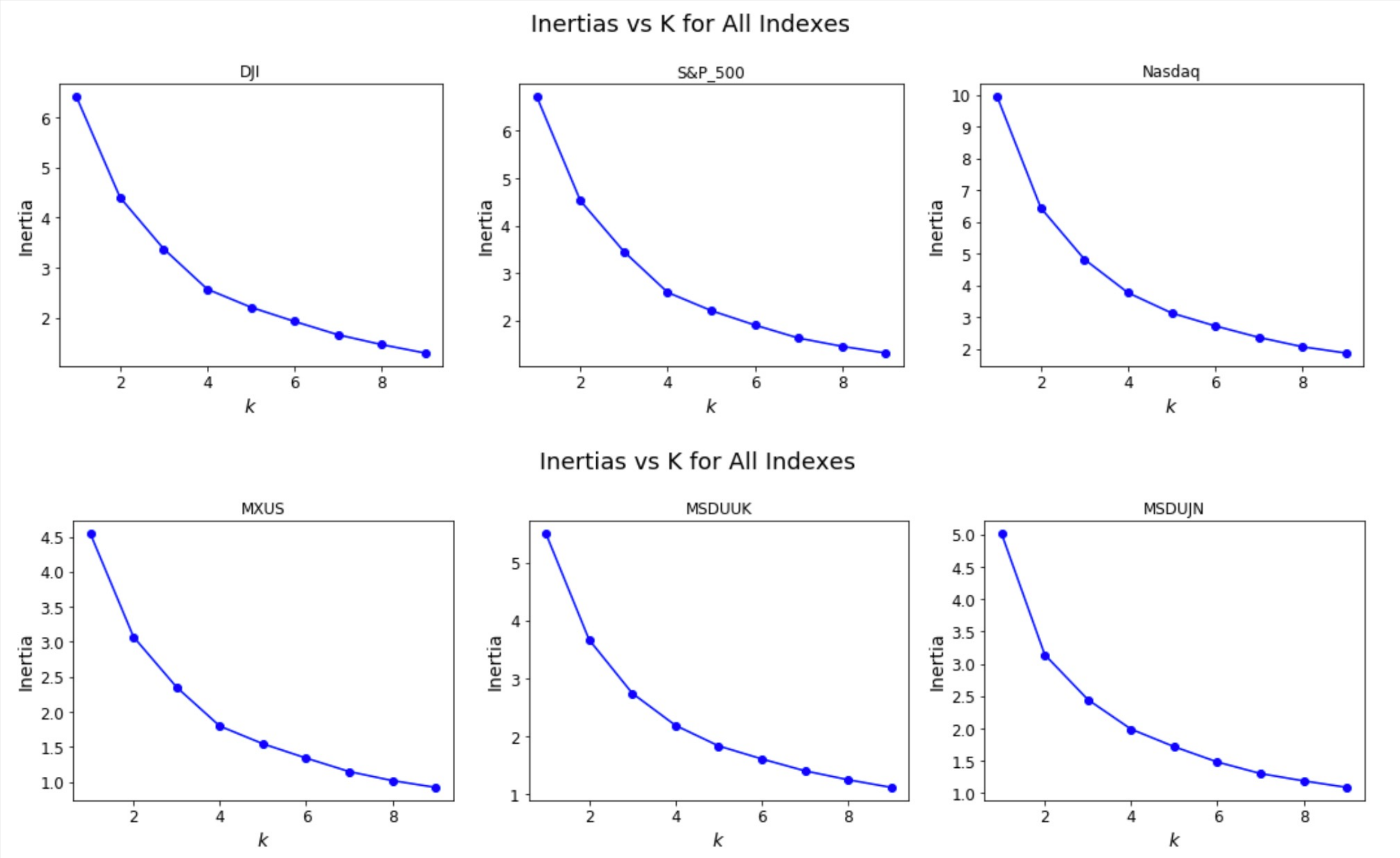
- *Clustering Classifier: k-Means*

- Method for selecting optimum # of clusters, Inertia Method
  - Plot of inertias vs. # clusters

*Refer to sections 6.1.8 Cluster Analysis with Unsupervised Learning*

# Empirical Results : *Cluster Analysis with Unsupervised Learning for Stock Index Classification*

- Plot Inertias vs K For All Indexes



# Empirical Results : *Cluster Analysis with Unsupervised Learning for Stock Index Classification*

- Centroids for all Indexes for K=4

### DJI Centroids

| | R_DJI | V_DJI | V_Ave_All_Indexes |
|---|---|---|---|
| **0** | -0.0492546 | 0.0941522 | 0.0983676 |
| **1** | -0.0235677 | 0.0265586 | 0.030316 |
| **2** | 0.00677049 | 0.0129709 | 0.0148565 |
| **3** | 0.0298591 | 0.0365303 | 0.042341 |

### S&P_500 Centroids

| | R_S&P_500 | V_S&P_500 | V_Ave_All_Indexes |
|---|---|---|---|
| **0** | -0.0546674 | 0.101313 | 0.101457 |
| **1** | -0.0245752 | 0.0291085 | 0.0316298 |
| **2** | 0.00691808 | 0.0135997 | 0.0149289 |
| **3** | 0.0330099 | 0.0399647 | 0.0431455 |

### Nasdaq Centroids

| | R_Nasdaq | V_Nasdaq | V_Ave_All_Indexes |
|---|---|---|---|
| **0** | -0.0757562 | 0.0810681 | 0.0660092 |
| **1** | -0.024924 | 0.0323096 | 0.0269447 |
| **2** | 0.0124658 | 0.0175666 | 0.0151691 |
| **3** | 0.0473658 | 0.0588896 | 0.0474487 |

### MXUS Centroids

| | R_MXUS | V_MXUS | V_Ave_All_Indexes |
|---|---|---|---|
| **0** | -0.0605353 | 0.0896649 | 0.0800013 |
| **1** | -0.0238612 | 0.0280177 | 0.0281822 |
| **2** | 0.0072913 | 0.0150212 | 0.0195132 |
| **3** | 0.0357966 | 0.0383795 | 0.0390106 |

### MSDUUK Centroids

| | R_MSDUUK | V_MSDUUK | V_Ave_All_Indexes |
|---|---|---|---|
| **0** | -0.0684867 | 0.0769637 | 0.0663507 |
| **1** | -0.0240379 | 0.0281672 | 0.0271886 |
| **2** | 0.00798854 | 0.0173268 | 0.0193995 |
| **3** | 0.0403084 | 0.0406009 | 0.0367707 |

### MSDUJN Centroids

| | R_MSDUJN | V_MSDUJN | V_Ave_All_Indexes |
|---|---|---|---|
| **0** | -0.0437838 | 0.0771822 | 0.0739504 |
| **1** | -0.0305874 | 0.0290289 | 0.0274579 |
| **2** | 0.00343401 | 0.0231735 | 0.0210083 |
| **3** | 0.0366376 | 0.0308732 | 0.0264983 |

# Conclusions

- The results presented by Refence 1 have been validated as demonstrated by comparison of Testing Sets' results. (*section 6.3.7.2* of Jupyter Notebook and Reference 1, Table 3)

- Derived financial *Engineered Network Parameters* were found NOT to improve forecasting results *(section 6.3.4.2.2 of Jupyter Notebook)*

- The daily log returns, five-day moving average return, five-day rolling volatility, average 5-day volatility between markets, and the engineered stock index classifications were determined to be the critical features in forecasting a market's next day price pattern

- The number of stock index classifications levels (S1, S2, S3, S4) derived from the Engineered Formulas in Reference 1 were validated via clustering analysis using K-Means.
  - For all stock indexes, the optimum number of clusters/classifications per index is four using the inertia method.
  - In all cases, all centroids have two positive 5-day average log returns, and two a negative ones, matching Ref 1 formula
  - Volatility values for each cluster centroids matches closely the logic use in Reference 1

- The results were determined to be applicable to different markets and time periods

- Four independent Machine Learning algorithms were utilized and provide forecasting accuracy between 75 to 82%

- Inspection of the Precision and Recall metrics and Confusion Matrix for each classifier further demonstrate the validity of the results.

- Price Transition patterns for Analyzed Markets with respect to Indexes' Classification further validate the results.
  - Transition's frequencies from one classification level to another were determined to be lower than the accuracy rate of the classifiers.

# Recommendations

- Evaluate the impact of different rolling volatility windows on forecasting results, that is: 20-day, 10-day, and 3-day

- Since the Derived financial *Engineered Network Parameters* were found NOT to improve forecasting results, evaluate if longer sliding windows can captured additional forecasting knowledge.
  - Evaluate 60, 90, 120, and 200 days sliding windows

- Evaluate Ensemble Methods

- Evaluate Deep Neural Networks
  - Classification NN
  - CNN
  - RNN ( WaveNet, others. Try forecasting a set, and then retrain and forecast one by one the testing set)

- Define a set of trading strategies and evaluate each in relation to the forecasting predictions
  - Across different sets of indexes, time periods

- Evaluate applicability of the forecasting method to different sectors within a market, and/or sectors across markets

- Use derived index classifications via unsupervised learning for forecasting next day price movement for each stock index
  - Evaluate clusters with 4, 5, and 6 different levels
  - Compare results to engineered index classifications & against trading strategies