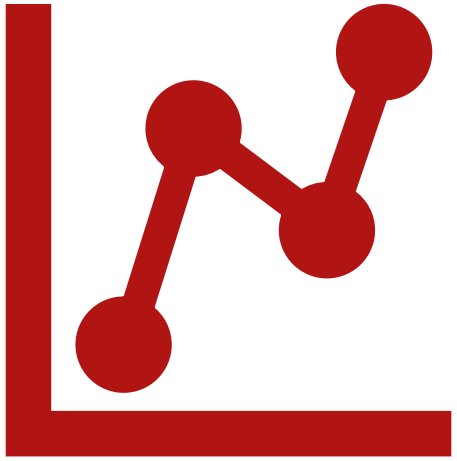


# Predicting Stock Trends With K-Nearest Neighbor

JASON ELLERBECK

ITCS – 5156

28 APRIL 2023

# Motivation

- Create an application can that predict the stock market
- Uses easy-to-understand understand data
- Apply the content we learned to real world data and use cases

# Problems & Challenges

- Limited understanding of stock trading
- Data collection
  - Getting features
- Unlike traditional programming
  - Output can be ambiguous
  - No code checks

# Main Paper

- Stock Market Trend Prediction using Supervised Learning, 2019
  - Asad Masood Khattak, Habib Ullah, Hassan Ali Khalid, Ammara Habib, Muhammad Zubair Asghar, and Fazal Masud Kundi

- K-Nearest Neighbor model to predict stock trend using easy to understand features
  - Data Collection
  - Data Normalization
  - Applying KNN

# Related Approaches

- A lot of research has gone into using machine learning for stock market investing
  - Requires an understanding of finical data and application

- Even though I could not replicate a lot of these studies due to lack of knowledge in finance, machine learning concepts could still be applied to this project

# Supporting Paper #1

- Application of Machine Learning in Stock Selection, 2022
  - Pengfei Li, Jungang Xu and, Mohammad Al-Hamami

- Comparison of different models predicting stocks
  - Feature Selection
  - How to compare model

# Supporting Paper #2

- A comparative study of supervised machine learning algorithms for stock market trend prediction, 2018
  - Indu Kumar, Kiran Dogra, Chetna Utreja, and Premlata Yadav

- Step-by-Step
- Uses high level data
- Highlights important metrics to evaluate model

# Replicating KNN Model

# Data Collection

- https://www.ksestocks.com/QuotationsData
  - 8 years of daily data
  - Data going back to September 2022
- Nestle Csv file:
  - Stock Ticker – Symbol
  - Date
  - Open price
  - Daily High
  - Daily Low
  - Close Price
  - Volume Traded

| | Symbol | Date | Open | High | Low | Close | Volume |
|---|---|---|---|---|---|---|---|
| **0** | NESTLE | 1-Sep-22 | 5780.00 | 5800.00 | 5711.0 | 5729.00 | 200 |
| **1** | NESTLE | 2-Sep-22 | 5638.13 | 5764.99 | 5620.0 | 5764.99 | 160 |
| **2** | NESTLE | 7-Sep-22 | 5760.00 | 5760.00 | 5760.0 | 5760.00 | 20 |
| **3** | NESTLE | 8-Sep-22 | 5760.00 | 5760.00 | 5760.0 | 5760.00 | 20 |
| **4** | NESTLE | 9-Sep-22 | 5760.00 | 5760.00 | 5760.0 | 5760.00 | 40 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **114** | NESTLE | 31-Mar-23 | 5280.00 | 5280.00 | 5015.0 | 5017.50 | 60 |
| **115** | NESTLE | 4-Apr-23 | 5390.90 | 5390.90 | 4887.1 | 5115.00 | 60 |
| **116** | NESTLE | 5-Apr-23 | 4950.10 | 5125.00 | 4950.1 | 5125.00 | 360 |
| **117** | NESTLE | 6-Apr-23 | 5200.00 | 5225.00 | 5030.0 | 5198.20 | 460 |
| **118** | NESTLE | 7-Apr-23 | 5198.20 | 5199.00 | 5005.0 | 5071.33 | 80 |

119 rows × 7 columns

# Data Normalization

- Target Feature:
  - Stock_return [1]

Stock_return = (current_day_closing_price – closing_price_of_previous_day) / previous_day_closing_price

- Day to day if a stock went up or down

# Data Normalization

- Computation Z – Score:

$$Z - score = Open - \frac{Mean\ of\ Open}{Stdv\ of\ Open}$$

[1]

# Data Normalization

- Identifying Outliers:
  - IF( Z-Score > 1 OR Z-Score < -1)  [1]

    mark as outlier

- Concrete data for the model

| | Symbol | Date | Open | High | Low | Close | Volume | Z-Score | Stock Return | isOutlier |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | NESTLE | 2-Sep-22 | 5638.13 | 5764.99 | 5620.0 | 5764.99 | 160 | 0.336149 | 0.006282 | 0 |
| **2** | NESTLE | 7-Sep-22 | 5760.00 | 5760.00 | 5760.0 | 5760.00 | 20 | 0.737363 | -0.000866 | 0 |
| 3 | NESTLE | 8-Sep-22 | 5760.00 | 5760.00 | 5760.0 | 5760.00 | 20 | 0.737363 | 0.000000 | 0 |
| **4** | NESTLE | 9-Sep-22 | 5760.00 | 5760.00 | 5760.0 | 5760.00 | 40 | 0.737363 | 0.000000 | 0 |
| **5** | NESTLE | 13-Sep-22 | 5760.00 | 5760.00 | 5760.0 | 5760.00 | 20 | 0.737363 | 0.000000 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **107** | NESTLE | 21-Mar-23 | 5490.00 | 5490.00 | 4752.0 | 5181.00 | 60 | -0.151516 | 0.012903 | 0 |
| **109** | NESTLE | 24-Mar-23 | 5299.99 | 5300.00 | 5100.1 | 5100.10 | 240 | -0.777056 | 0.019989 | 0 |
| **111** | NESTLE | 28-Mar-23 | 5300.00 | 5300.00 | 5200.1 | 5233.70 | 60 | -0.777023 | 0.015257 | 0 |
| **114** | NESTLE | 31-Mar-23 | 5280.00 | 5280.00 | 5015.0 | 5017.50 | 60 | -0.842866 | 0.008056 | 0 |
| **115** | NESTLE | 4-Apr-23 | 5390.90 | 5390.90 | 4887.1 | 5115.00 | 60 | -0.477767 | 0.019432 | 0 |

# Data Normalization

- 72 rows of data
- Target:
  - Stock_return
- Features:
  - Open Price
  - Daily High
  - Daily Low
  - Close Price
- SciKitLearn - train_test_split (80/20)

# Applying KNN

- SciKitLearn – KNeighborsRegressor

| Parameter | Description |
|---|---|
| Algorithm = 'auto' | Will attempt to decide the most appropriate algorithm based on the values passed to fit method. |
| Leaf size = 30 (default = 30) | This can affect the speed of the construction and query, as well as the memory required to store the tree. The optimal value depends on the nature of the problem. |
| Metric = 'minkowski' | The distance metric to be used for the tree. The default metric is minkowski. |
| Metric prams = None | Additional keyword arguments for the metric function. |
| n-Jobs = 1 | The number of parallel jobs to run for neighbors' search. |
| n-neighbors = n | Number of neighbors to be used by default for kneighbors queries. |
| p = 2 | For arbitrary p, minkowski_distance (l_p) is used. |
| weight = 'uniform' | All points in each neighborhood are weighted equally. |

[1]

# Evaluating Results

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- Compare error of normalized data and unnormalized data

# Provided Results

**Table 6: Prediction results on raw dataset**

| Iteration | RMSE | MAE | Logcosh |
|---|---|---|---|
| 5 | 202955735.8 | 11976.81202 | Inf |
| 11 | 188881104.5 | 11746.46341 | Inf |
| 17 | 180075597.2 | 11570.63793 | Inf |
| 23 | 174864623.6 | 11430.98562 | Inf |
| 29 | 173025251.2 | 11378.50613 | Inf |
| 35 | 171312709.5 | 11335.03967 | Inf |
| 41 | 169612906.4 | 11275.85507 | Inf |

[1]

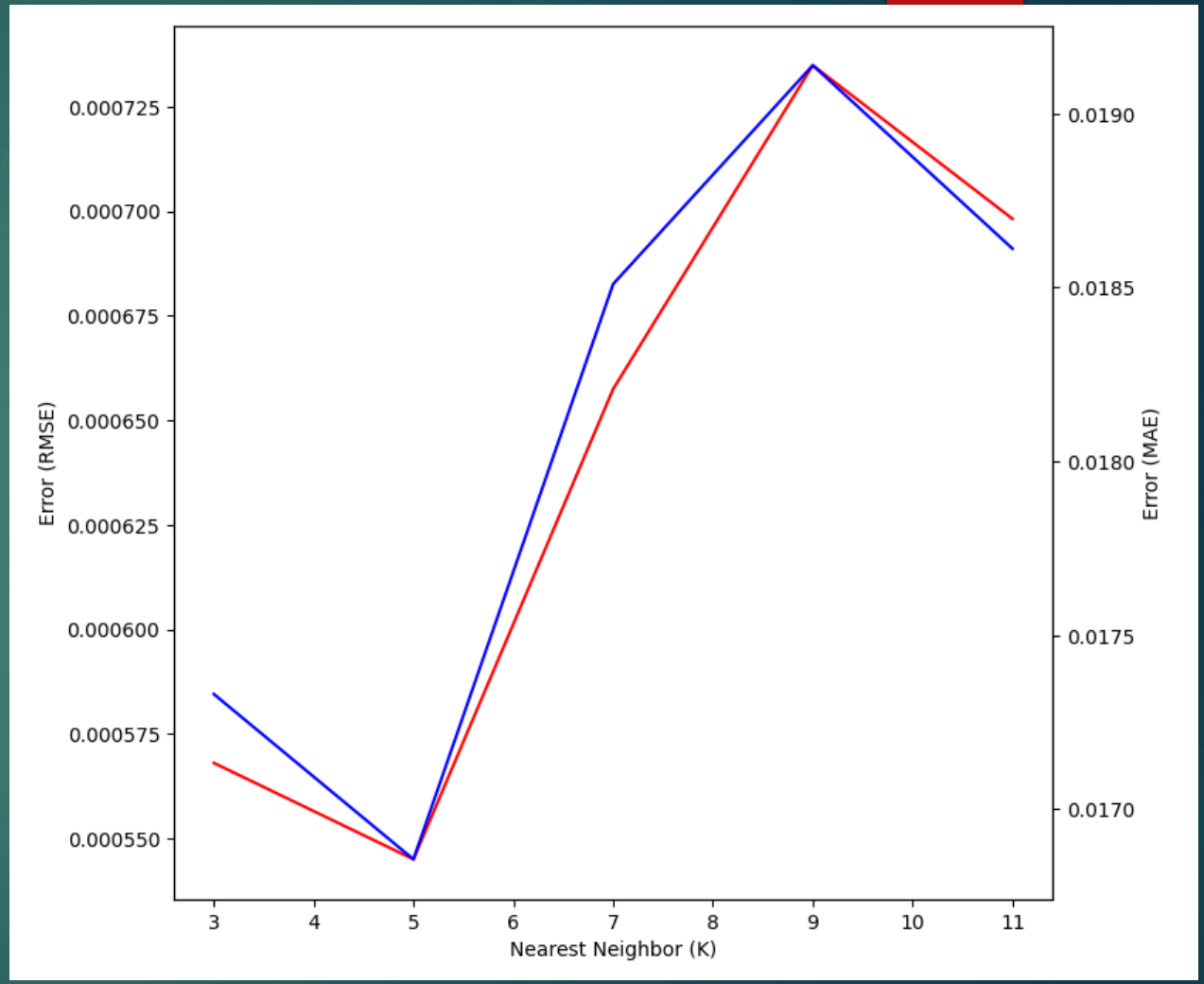
**Table 8: A sample listing of normalized dataset result**

| Iteration | RMSE | MAE | Logcosh |
|---|---|---|---|
| 5 | 4.42E-05 | 0.004892914 | 0.003716503 |
| 11 | 6.07E-05 | 0.005722075 | 0.005097995 |
| 17 | 6.31E-05 | 0.005944986 | 0.005303328 |
| 23 | 6.64E-05 | 0.006160232 | 0.005573119 |
| 29 | 6.71E-05 | 0.006216439 | 0.005634309 |
| 35 | 6.94E-05 | 0.006315764 | 0.005827321 |
| 41 | 7.05E-05 | 0.006379043 | 0.005918586 |

[1]

# My Results

| | RSME | MAE | Iteration |
|---|---|---|---|
| 0 | 0.000568 | 0.017331 | 3 |
| 1 | 0.000545 | 0.016856 | 5 |
| 2 | 0.000657 | 0.018510 | 7 |
| 3 | 0.000735 | 0.019139 | 9 |
| 4 | 0.000698 | 0.018612 | 11 |



# Conclusion and Future Work

- Lessons learned:
  - Data collection
  - Formatting data
- Additional Models
  - SVM
  - Random Forest
  - Naïve Bayes
- Additional Metrics
  - Accuracy
  - F - Measure
- Comparing the models

# References


[1] Asad Masood Khattak, Habib Ullah, Hassan Ali Khalid, Ammara Habib, Muhammad Zubair Asghar, and Fazal Masud Kundi. 2019. Stock Market Trend Prediction using Supervised Learning. In Proceedings of the 10th International Symposium on Information and Communication Technology (SoICT '19). Association for Computing Machinery, New York, NY, USA, 85–91. https://doi.org/10.1145/3368926.3369680

[2] Li, Pengfei, Xu, Jungang and AI-Hamami, Mohammad. "Application of machine learning in stock selection" Applied Mathematics and Nonlinear Sciences, vol.0, no.0, 2022, pp.-. https://doi.org/10.2478/amns.2022.1.00025

[3] I. Kumar, K. Dogra, C. Utreja and P. Yadav, "A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018, pp. 1003-1007, doi: 10.1109/ICICCT.2018.8473214.