A Machine Learning Strategy for Optimal Stock Trading

Sayan Das
dassayan0013@gmail.com

Alex Nirmal Minj
alexnm.math.ug@jadavpuruniversity.in

Agnik Dasgupta
agnikd76@gmail.com

Debarpan Ghosh
dghosh4527@gmail.com

October 30, 2024

Contents

After experimenting with various models and techniques, the framework for this final project was laid down by Sayan, which was then further improvised, refined and fine-tuned by Alex.

Debarpan and Agnik did the tedious work of researching the various datasets and selecting them for training the model, since getting complete usable NIFTY data was next to impossible, we settled on S&P500 data.

We would also like to thank our mentor, Dr. Snehalika Lall, for her invaluable insight.

1 Introduction

1.1 Aim:

We aim to use an unsupervised machine learning strategy to create a portfolio that outperforms the SPDR S&P 500 (SPY) exchange-traded fund eventually.

1.2 Datasets and API used:

- List of current S&P500 stocks from Wikipedia (as of October 2024)
- Yahoo Finance API

1.3 Github repository and instructions for running the code

The model was coded using Python 3 in a Jupyter notebook. It is available at this link. To run the code given in the `model.ipynb` file, it is of course necessary to have the latest version of Python and JupyterLab Notebook installed. To start the JupyterLab Notebook, open your terminal within the working directory containing the `model.ipynb` file. Within the terminal run the following command:

```
py -m notebook
```

The JupyterLab notebook environment should open up in your browser. Before opening the `model.ipynb` code, go to `File> New> Terminal` and install the requisite packages:

```
py -m !pip install pandas pandas_ta pandas_datareader numpy matplotlib
py -m !pip install statsmodels datetime yfinance scikit-learn PyPortfolioOpt
```

1.4 Packages used:

- pandas,
- pandas_ta,
- pandas_datareader,
- numpy,
- matplotlib,
- statsmodels,
- datetime,
- yfinance,
- scikit-learn,
- PyPortfolioOpt

2 Methodology

We now discuss the methodology used.

2.1 Features and technical indicators to calculate for each stock:

- Garman-Klass volatility

  Defined as
  $$r_t = \frac{(\ln H_t - \ln L_t)^2}{2} - 2(\ln 2 - 1)(\ln C_t - \ln O_t)^2,$$
  where O_t, C_t, H_t, L_t denote the opening, closing, high, low prices of day t respectively.

- Relative strength index (RSI)

  Relative strength:
  $$\text{RS} = \frac{\text{Average gain}}{\text{Average loss}}.$$
  Relative strength index:
  $$\text{RSI} = 100 - \frac{100}{1 + \text{RS}}.$$

- Bollinger bands

  $$\text{Upper band} = \text{MA}(n) + k\sigma(n),$$
  $$\text{Middle band} = \text{MA}(n),$$
  $$\text{Lower band} = \text{MA}(n) - k\sigma(n),$$
  where $\text{MA}(n)$ is the moving average over n periods and $\sigma(n)$ is the standard deviation over n periods. Usually we take $n = 20$ and $k = 2$.

- Average true range (ATR)

  Given by
  $$\frac{1}{n}\sum_{t=1}^{n} TR(t),$$
  where n is the number of periods, and $TR(t)$ is the true range of day t defined by
  $$TR(t) = \max((H_t - L_t), |H_t - C_{t-1}|, |L_t - C_{t-1}|).$$

- Moving average convergence divergence (MACD)

  Given by $\text{MACD} = \text{EMA}_{12} - \text{EMA}_{26}$, where EMA_t denotes the exponential moving average over t periods.

- Dollar volume liquidity

  Dollar volume liquidity is the price of a stock multiplied by its daily trading volume

2.2 Fama-French factors and stock filtering:

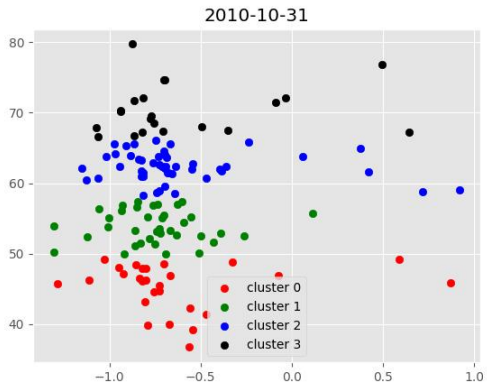
To reduce the model's training time we aggregate the data on a monthly level, filtering the top 150 most liquid stocks per month. We calculate 5-year moving average of dollar volume of each stock before filtering. We then compute monthly returns for different time horizons as features. We download the five Fama-French factors, namely 1. market risk, 2. size, 3. value, 4. operating profitability, and 5. investment

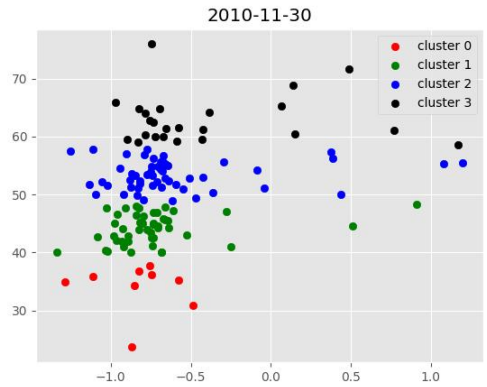
which have been empirically proven for asset returns and are used to assess the risk/return profile of portfolios. It is natural to include past factor exposures as financial features in our model.

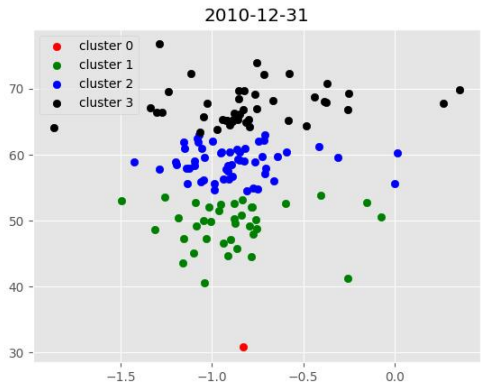
2.3 K-Means clustering and portfolio optimisation:

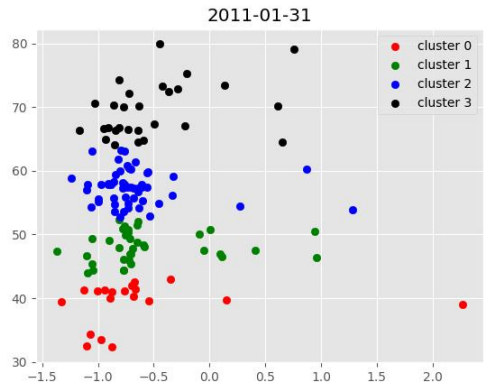
We filter out stocks with < 10 months of data, then calculate rolling factor betas and join it with the main features dataframe. Finally, we use an unsupervised K-Means clustering algorithm to group similar assets based on their features into 4 clusters. We have used the normalised values for all indicators except RSI, because we want to target specific RSI values. In fact our strategy is built on the hypothesis that momentum is persistent and that stocks clustered around RSI 70 centroid should continue to outperform in the following month - thus we select stocks corresponding to cluster 3.

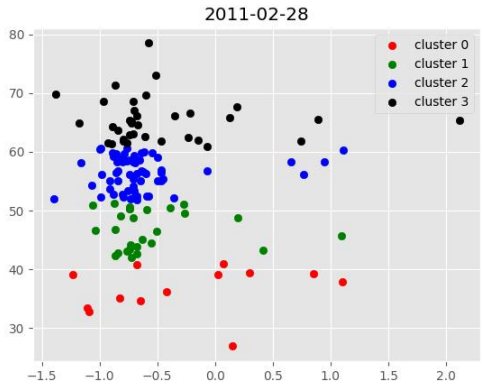
2.3.1 Cluster plots for each month

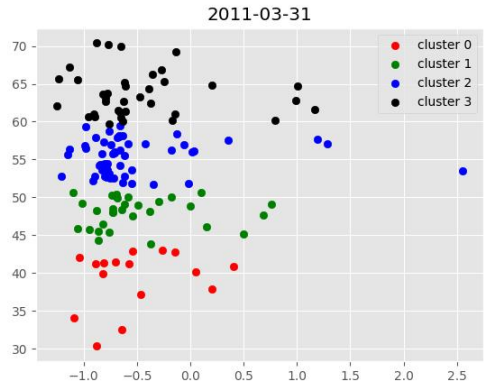


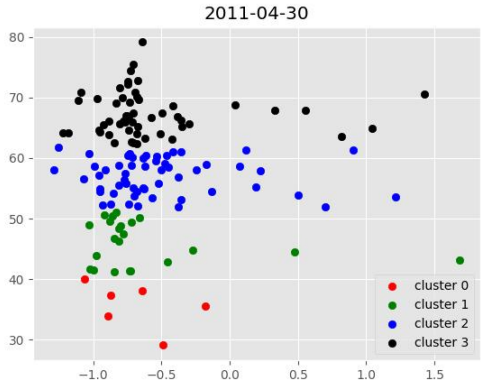


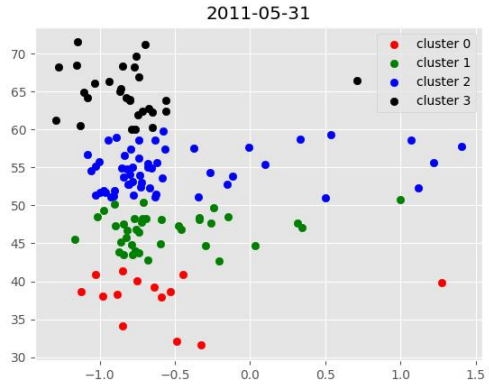


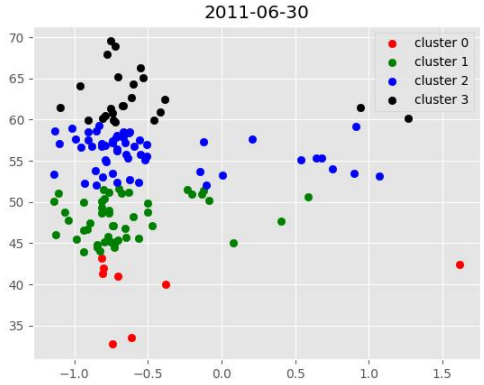


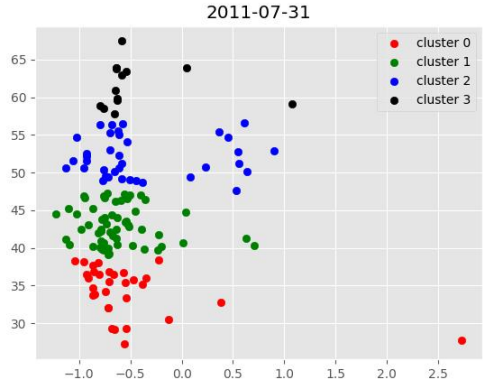


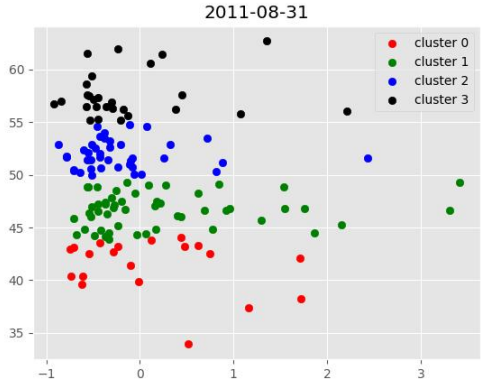


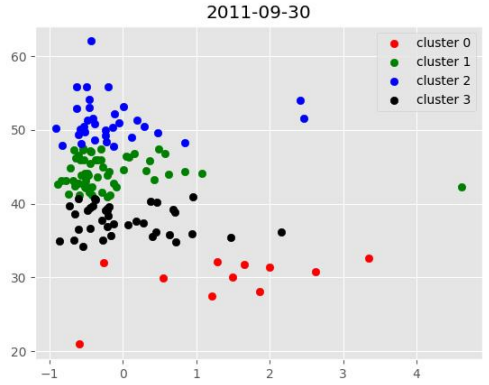

2011-04-30
cluster 0
cluster 1
cluster 2
cluster 3

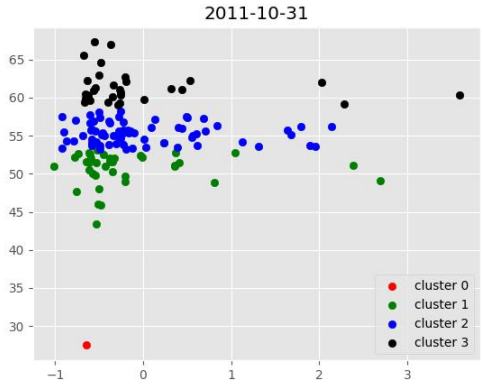

2011-05-31
cluster 0
cluster 1
cluster 2
cluster 3

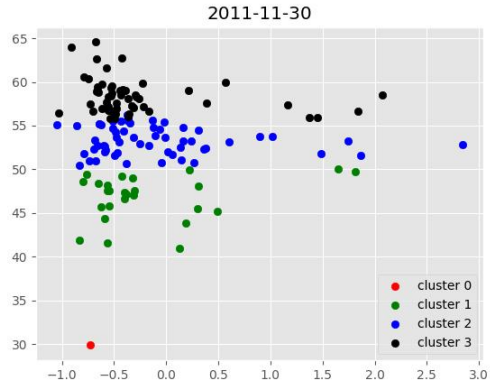

2011-06-30
cluster 0
cluster 1
cluster 2
cluster 3

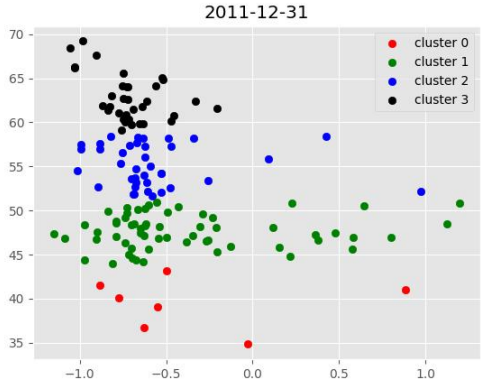

2011-07-31
cluster 0
cluster 1
cluster 2
cluster 3

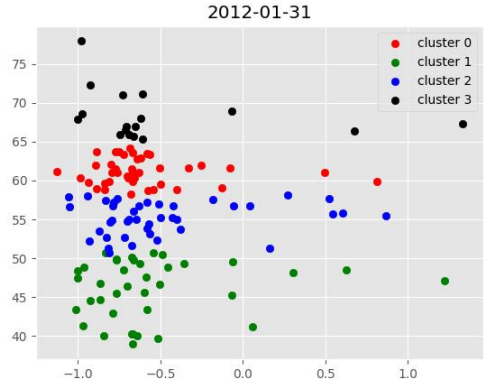

2011-08-31
cluster 0
cluster 1
cluster 2
cluster 3

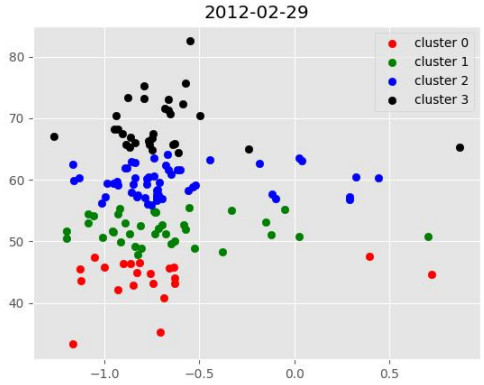

2011-09-30
cluster 0
cluster 1
cluster 2
cluster 3

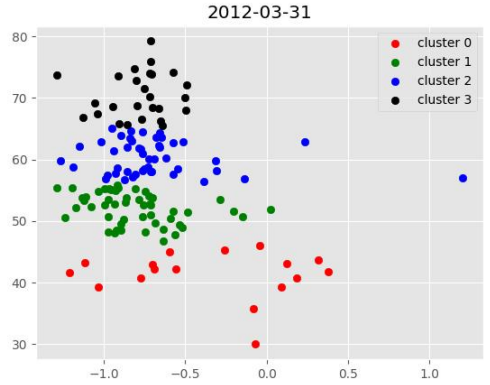

2011-10-31
cluster 0
cluster 1
cluster 2
cluster 3

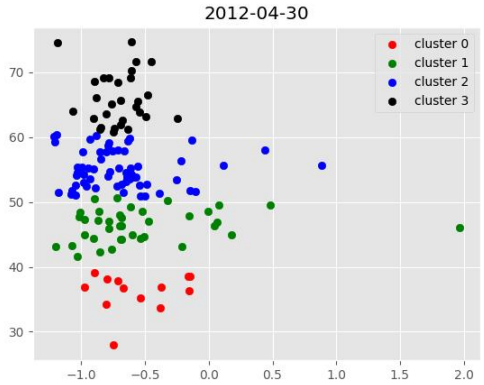

2011-11-30
cluster 0
cluster 1
cluster 2
cluster 3

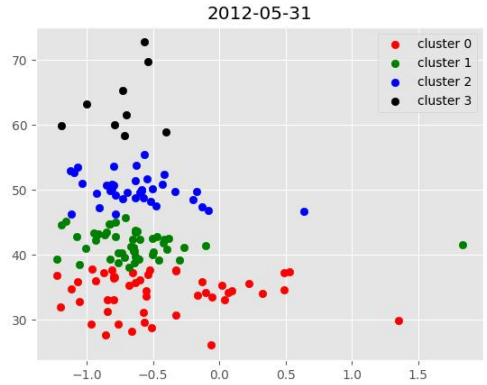

2011-12-31
cluster 0
cluster 1
cluster 2
cluster 3

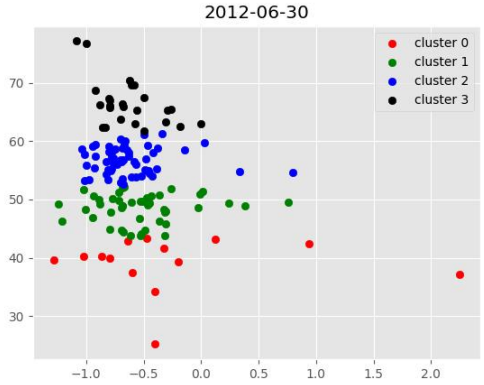

2012-01-31
cluster 0
cluster 1
cluster 2
cluster 3

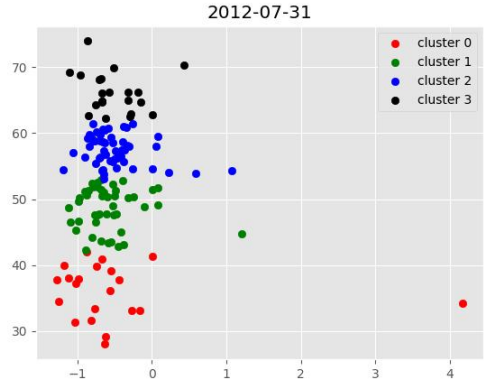

2012-02-29
cluster 0
cluster 1
cluster 2
cluster 3

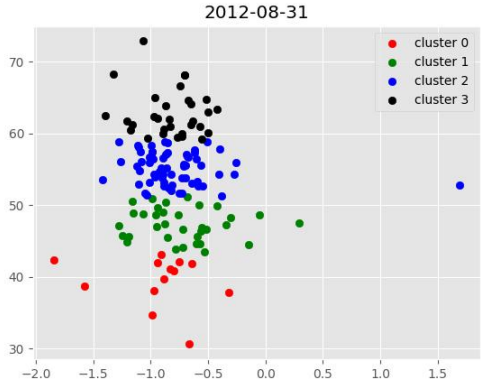

2012-03-31
cluster 0
cluster 1
cluster 2
cluster 3

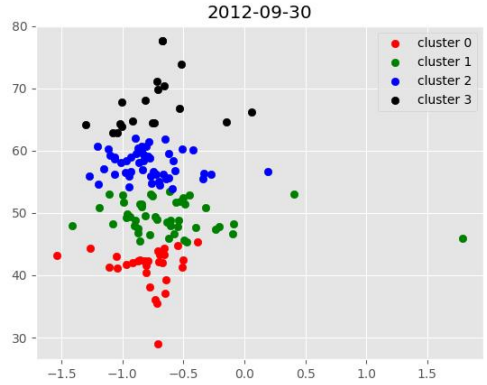

2012-04-30
cluster 0
cluster 1
cluster 2
cluster 3

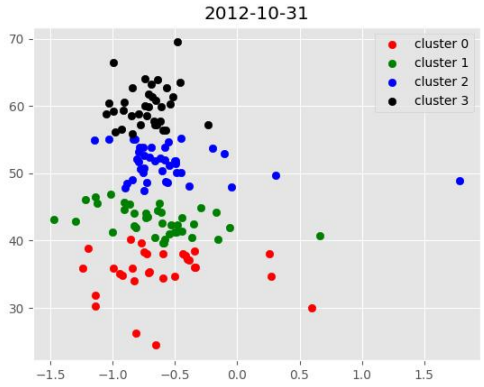

2012-05-31
cluster 0
cluster 1
cluster 2
cluster 3

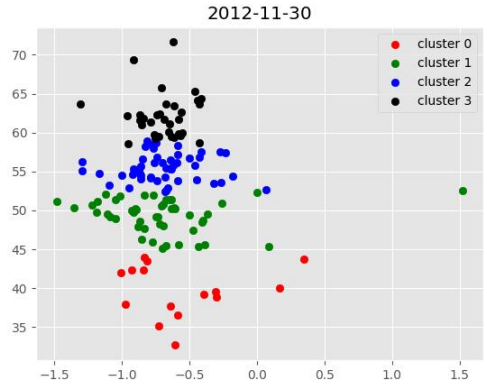

2012-06-30
cluster 0
cluster 1
cluster 2
cluster 3

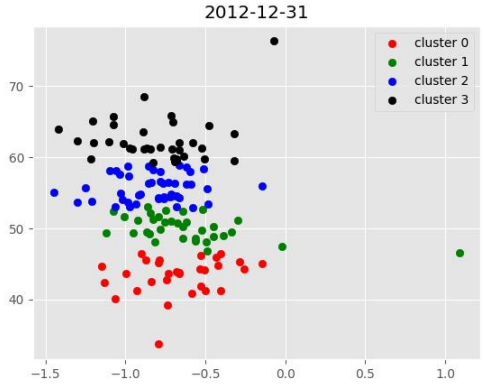

2012-07-31
cluster 0
cluster 1
cluster 2
cluster 3

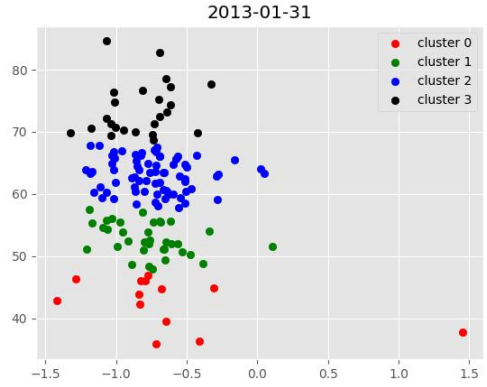

2012-08-31
cluster 0
cluster 1
cluster 2
cluster 3

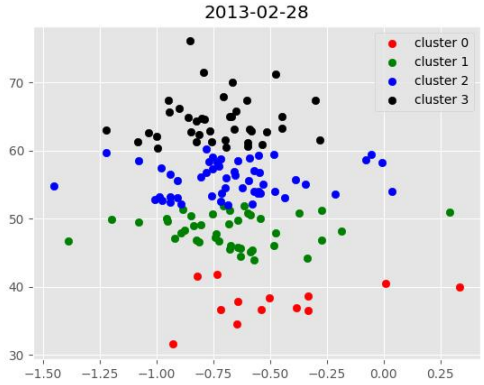

2012-09-30
cluster 0
cluster 1
cluster 2
cluster 3

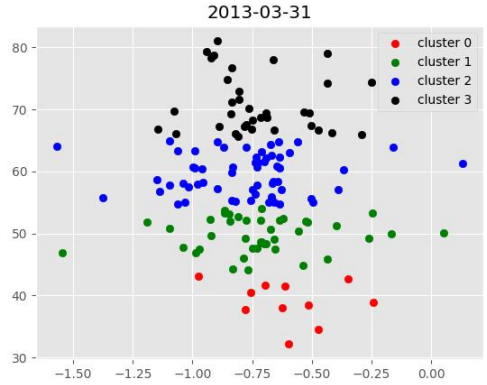

2012-10-31
cluster 0
cluster 1
cluster 2
cluster 3

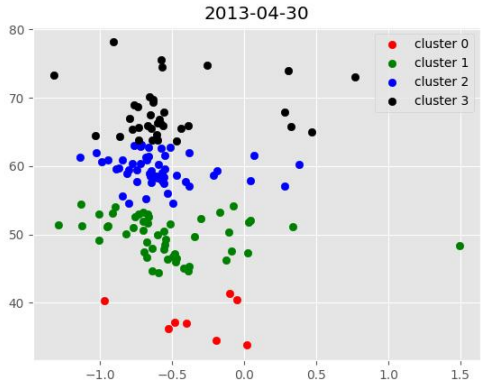

2012-11-30
cluster 0
cluster 1
cluster 2
cluster 3

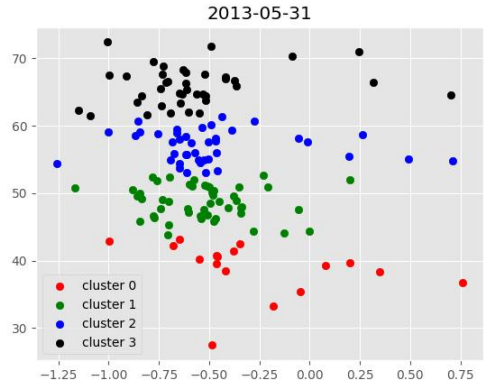

2012-12-31
cluster 0
cluster 1
cluster 2
cluster 3

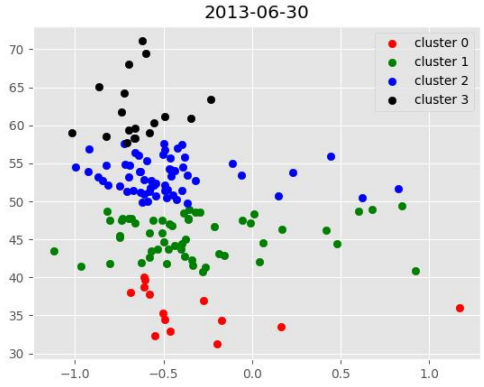

2013-01-31
cluster 0
cluster 1
cluster 2
cluster 3

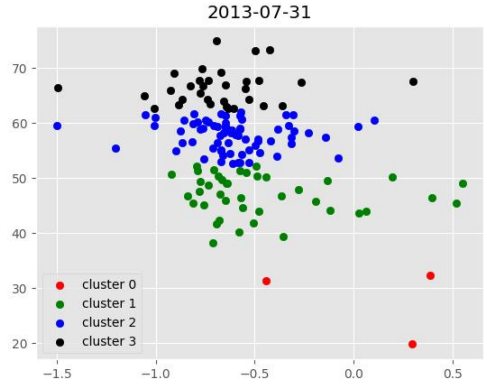

2013-02-28
cluster 0
cluster 1
cluster 2
cluster 3

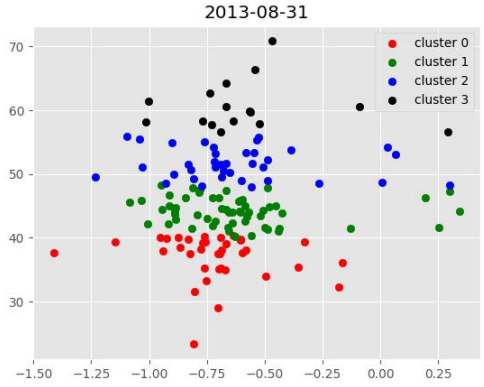

2013-03-31
cluster 0
cluster 1
cluster 2
cluster 3

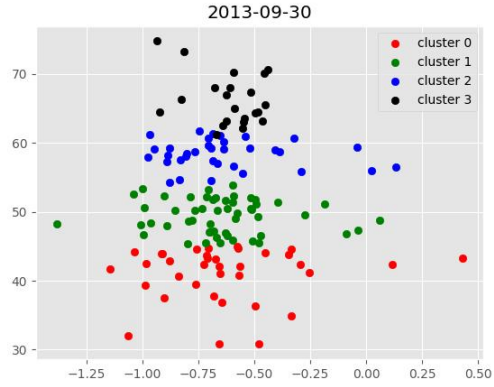

2013-04-30
cluster 0
cluster 1
cluster 2
cluster 3

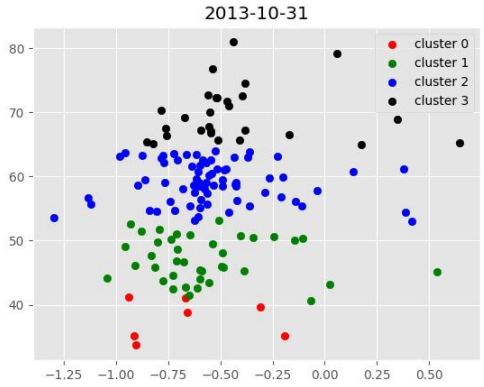

2013-05-31
cluster 0
cluster 1
cluster 2
cluster 3

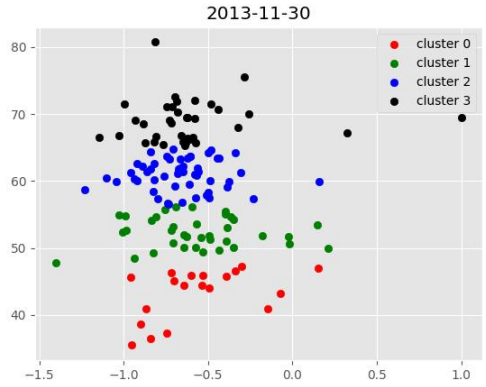

2013-06-30
cluster 0
cluster 1
cluster 2
cluster 3

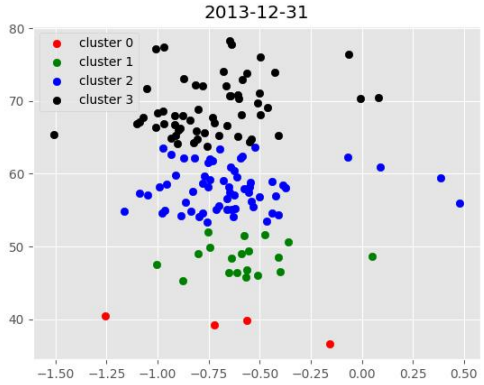

2013-07-31
cluster 0
cluster 1
cluster 2
cluster 3

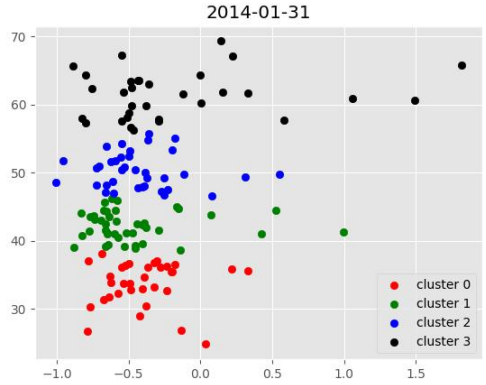

2013-08-31
cluster 0
cluster 1
cluster 2
cluster 3

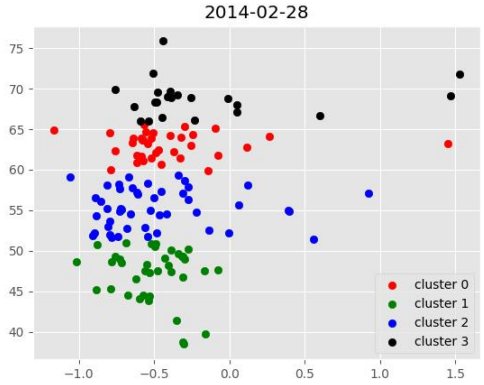

2013-09-30
cluster 0
cluster 1
cluster 2
cluster 3

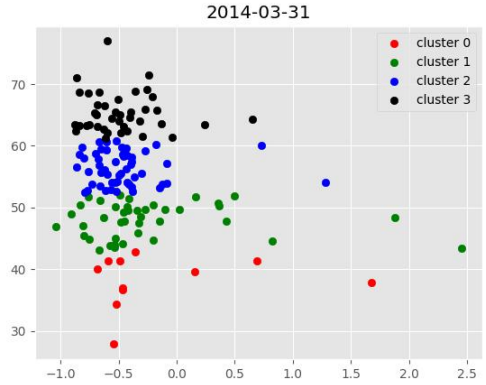

2013-10-31
cluster 0
cluster 1
cluster 2
cluster 3

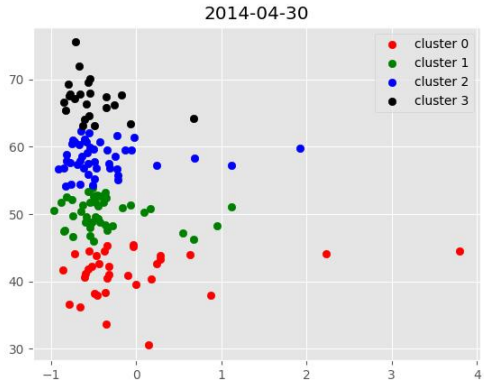

2013-11-30
cluster 0
cluster 1
cluster 2
cluster 3

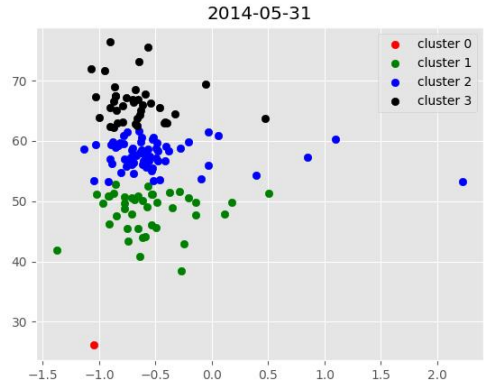

2013-12-31
cluster 0
cluster 1
cluster 2
cluster 3

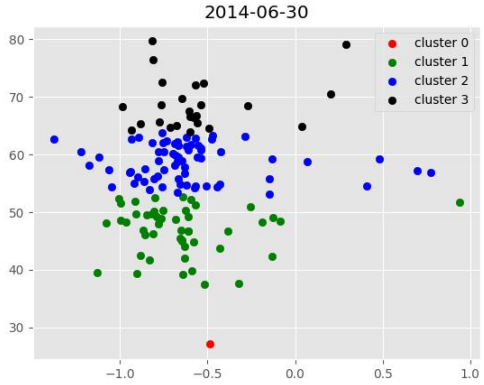

2014-01-31
cluster 0
cluster 1
cluster 2
cluster 3

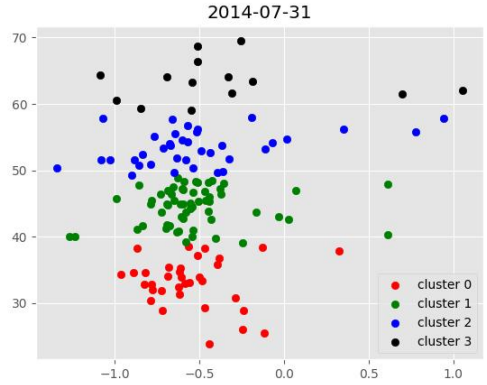

2014-02-28
cluster 0
cluster 1
cluster 2
cluster 3

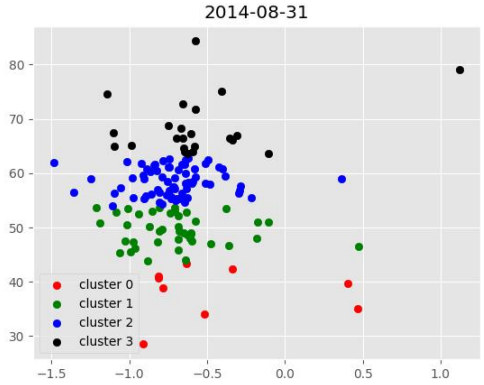

2014-03-31
cluster 0
cluster 1
cluster 2
cluster 3

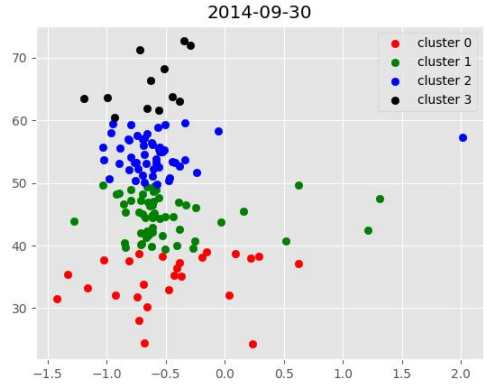

2014-04-30
cluster 0
cluster 1
cluster 2
cluster 3

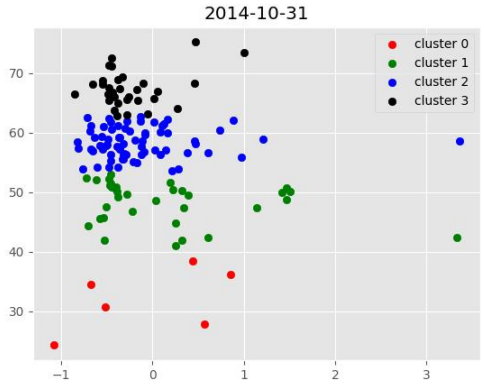

2014-05-31
cluster 0
cluster 1
cluster 2
cluster 3

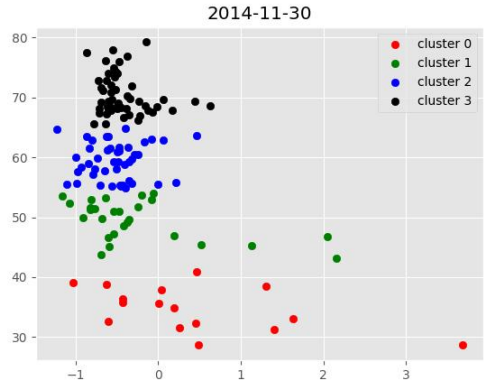

2014-06-30
cluster 0
cluster 1
cluster 2
cluster 3

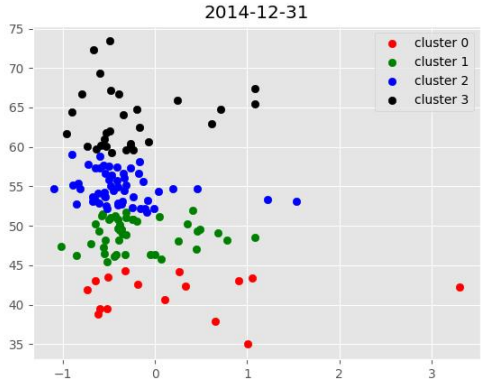

2014-07-31
cluster 0
cluster 1
cluster 2
cluster 3

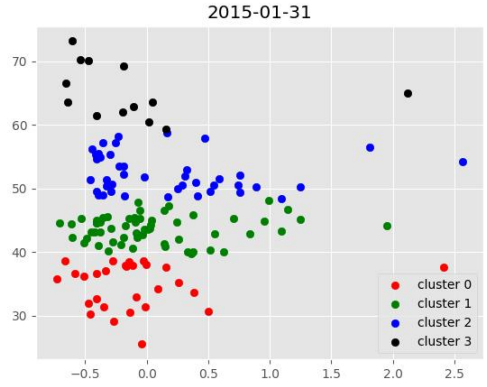

2014-08-31
cluster 0
cluster 1
cluster 2
cluster 3

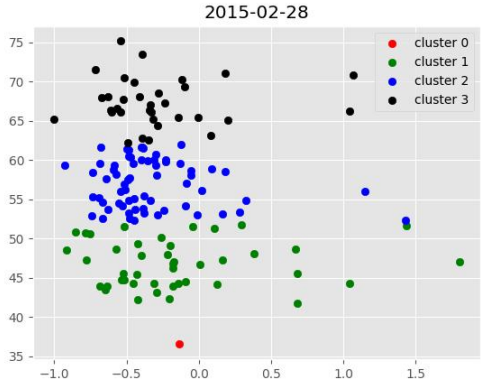

2014-09-30
cluster 0
cluster 1
cluster 2
cluster 3

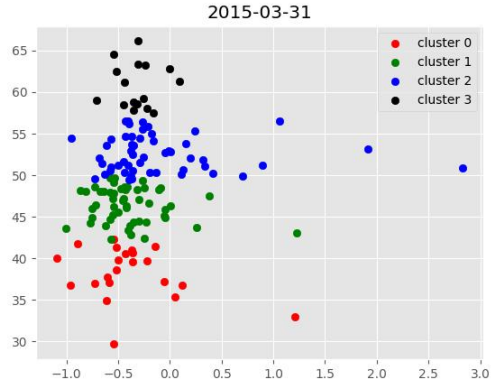

2014-10-31
cluster 0
cluster 1
cluster 2
cluster 3

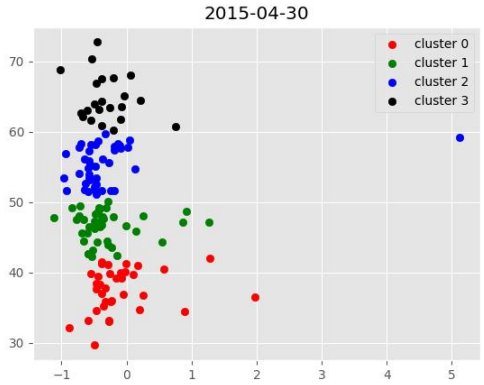

2014-11-30
cluster 0
cluster 1
cluster 2
cluster 3

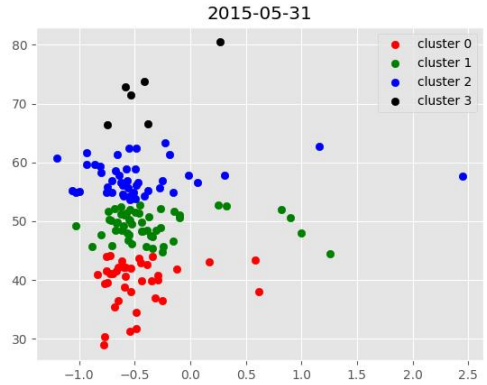

2014-12-31
cluster 0
cluster 1
cluster 2
cluster 3

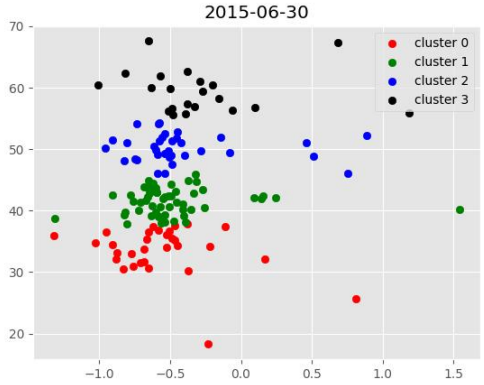

2015-01-31
cluster 0
cluster 1
cluster 2
cluster 3

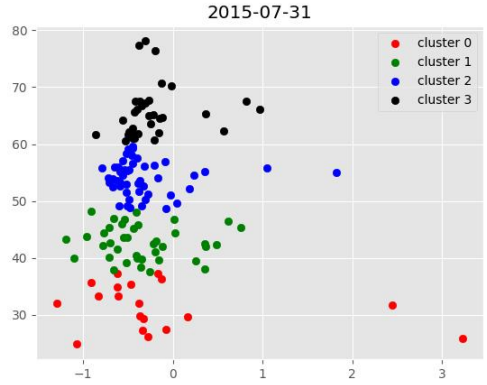

2015-02-28
cluster 0
cluster 1
cluster 2
cluster 3

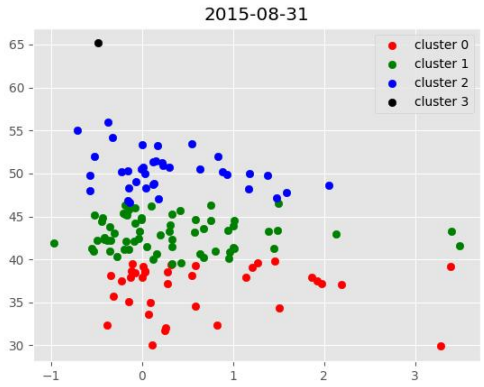

2015-03-31
cluster 0
cluster 1
cluster 2
cluster 3

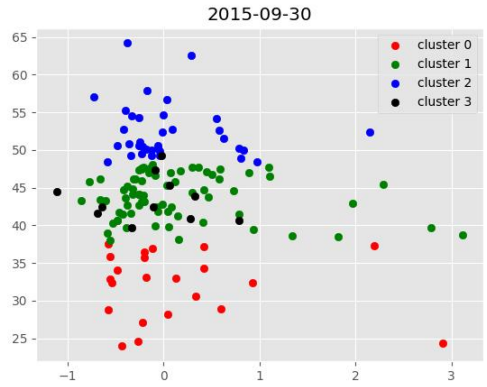

2015-04-30
cluster 0
cluster 1
cluster 2
cluster 3

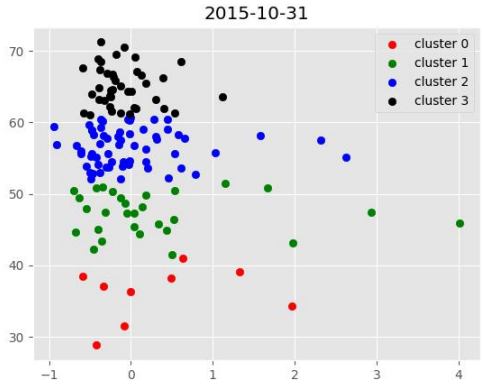

2015-05-31
cluster 0
cluster 1
cluster 2
cluster 3

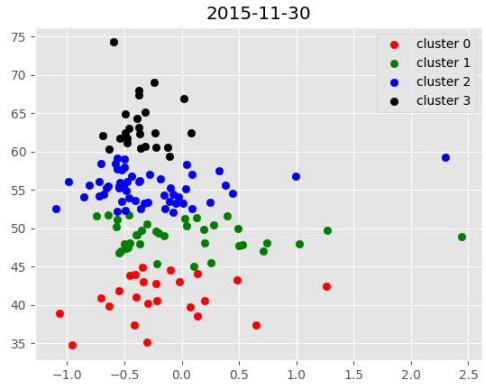

2015-06-30
cluster 0
cluster 1
cluster 2
cluster 3

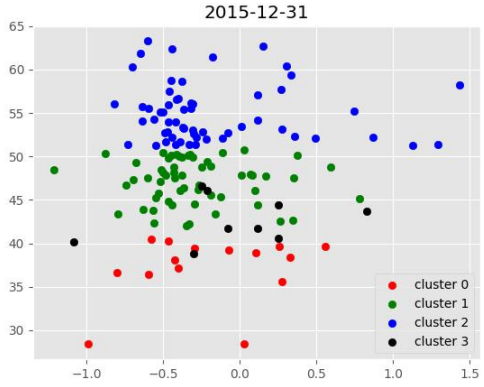

2015-07-31
cluster 0
cluster 1
cluster 2
cluster 3

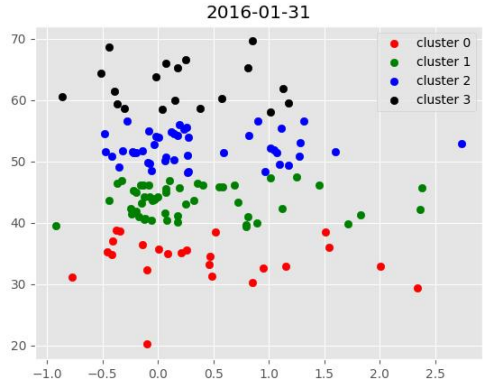

2015-08-31
cluster 0
cluster 1
cluster 2
cluster 3

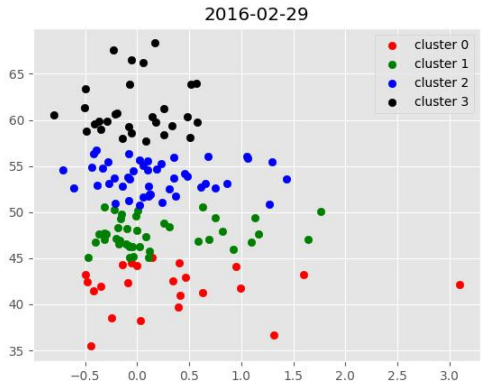

2015-09-30
cluster 0
cluster 1
cluster 2
cluster 3

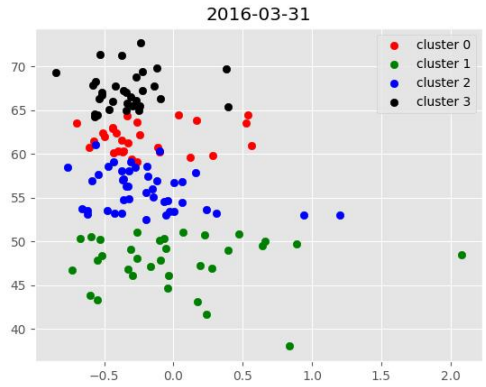

2015-10-31
cluster 0
cluster 1
cluster 2
cluster 3

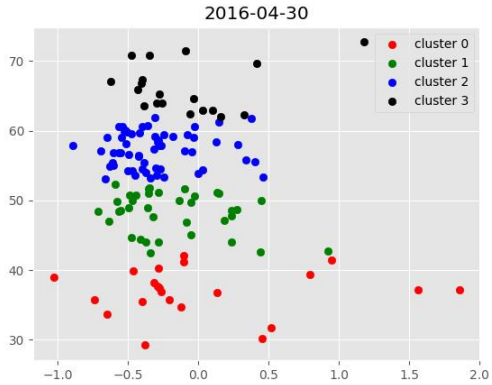

2015-11-30
cluster 0
cluster 1
cluster 2
cluster 3

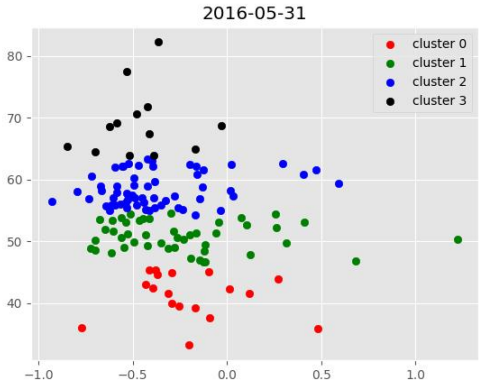

2015-12-31
cluster 0
cluster 1
cluster 2
cluster 3

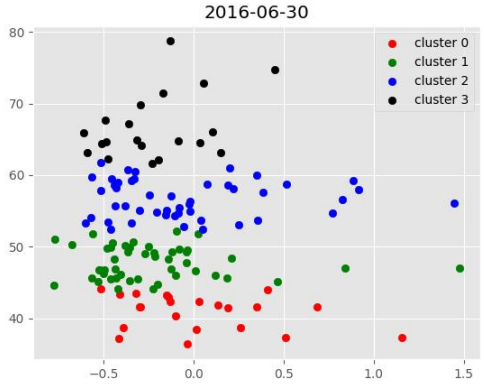

2016-01-31
cluster 0
cluster 1
cluster 2
cluster 3

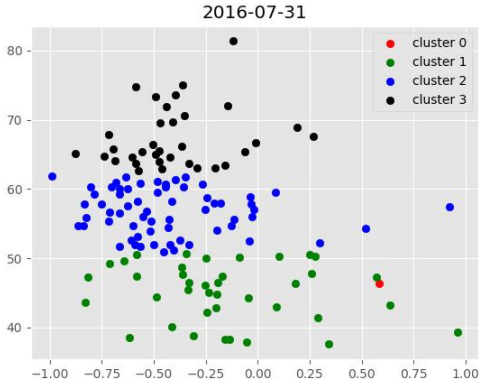

2016-02-29
cluster 0
cluster 1
cluster 2
cluster 3

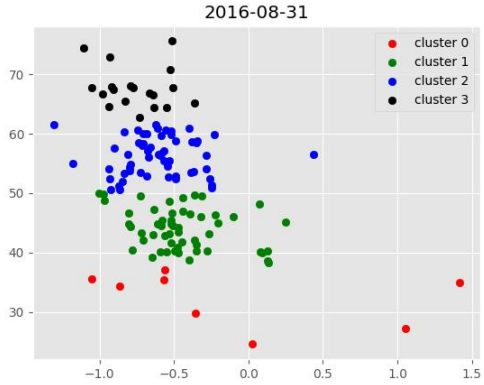

2016-03-31
cluster 0
cluster 1
cluster 2
cluster 3

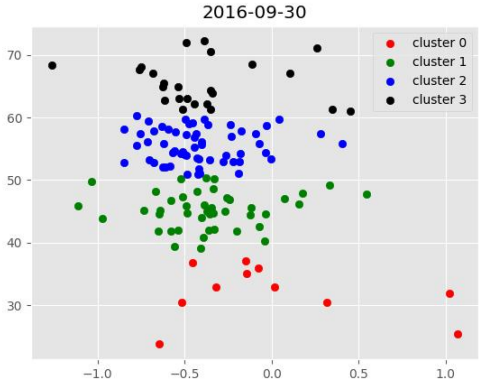

2016-04-30
cluster 0
cluster 1
cluster 2
cluster 3

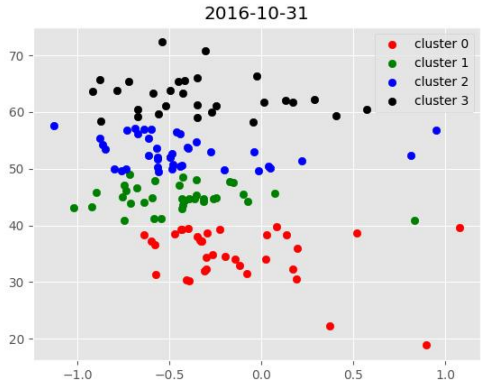

2016-05-31
cluster 0
cluster 1
cluster 2
cluster 3

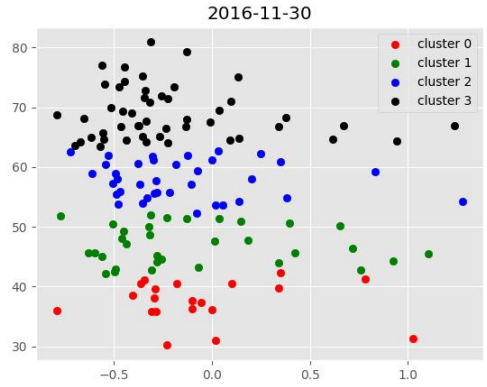

2016-06-30
cluster 0
cluster 1
cluster 2
cluster 3

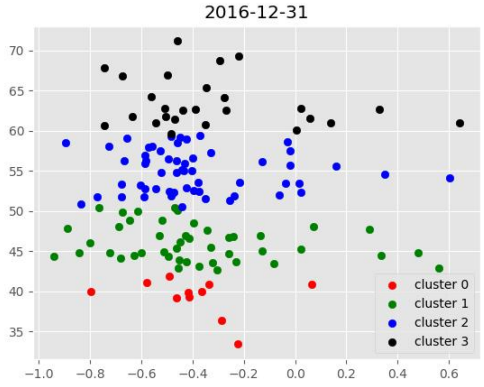

2016-07-31
cluster 0
cluster 1
cluster 2
cluster 3

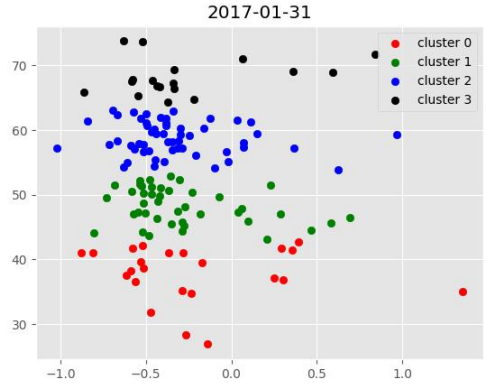

2016-08-31
cluster 0
cluster 1
cluster 2
cluster 3

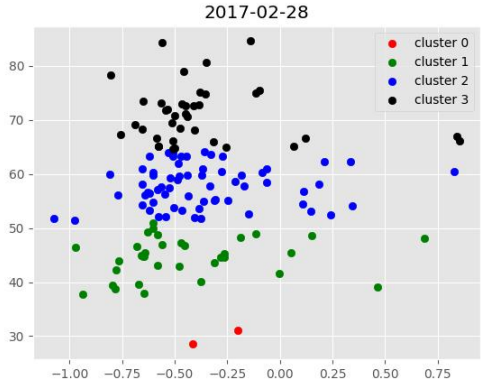

2016-09-30
cluster 0
cluster 1
cluster 2
cluster 3

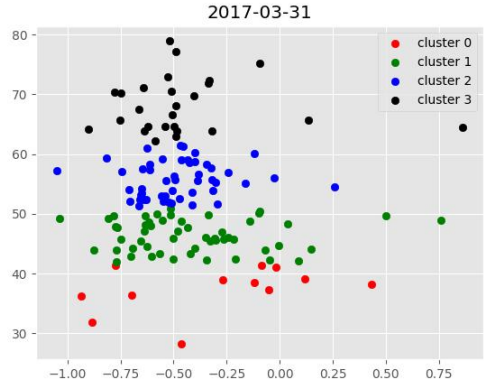

2016-10-31
cluster 0
cluster 1
cluster 2
cluster 3

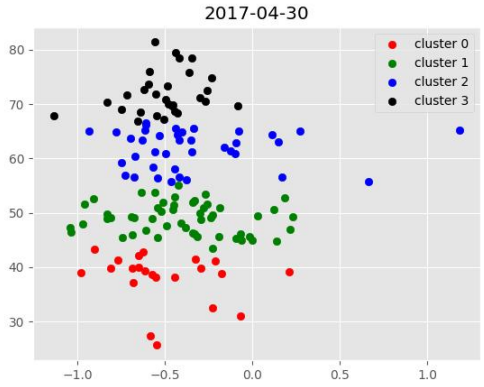

2016-11-30
cluster 0
cluster 1
cluster 2
cluster 3

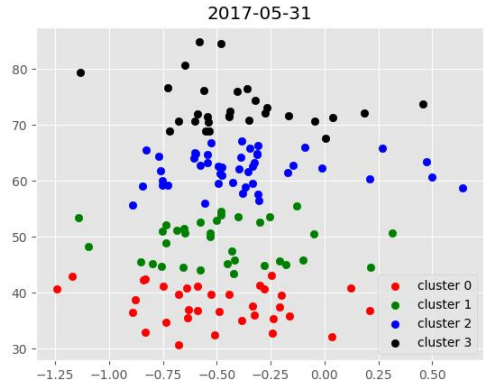

2016-12-31
cluster 0
cluster 1
cluster 2
cluster 3

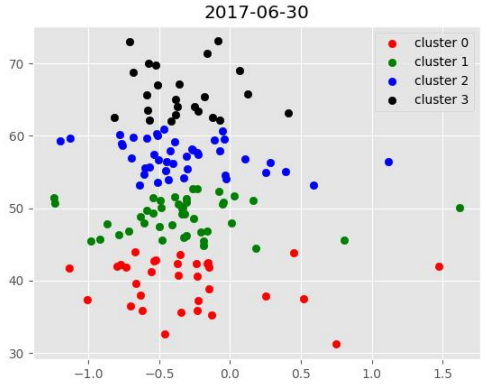

2017-01-31
cluster 0
cluster 1
cluster 2
cluster 3

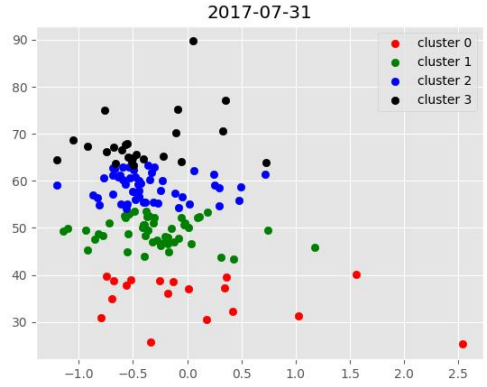

2017-02-28
cluster 0
cluster 1
cluster 2
cluster 3

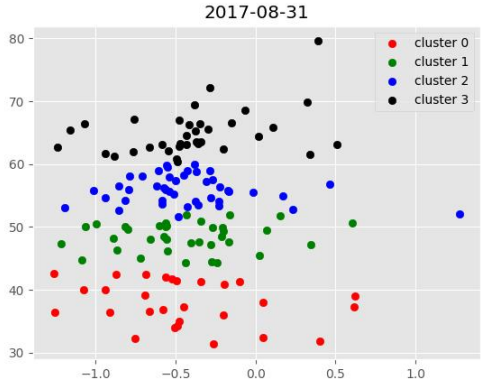

2017-03-31
cluster 0
cluster 1
cluster 2
cluster 3

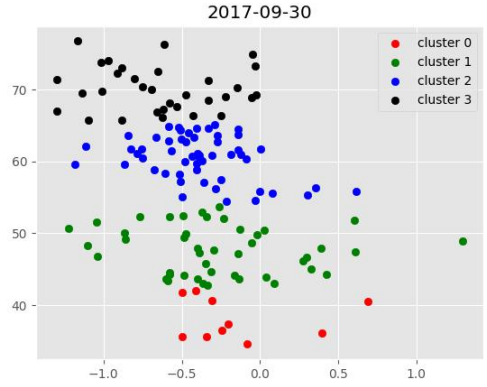

2017-04-30
cluster 0
cluster 1
cluster 2
cluster 3

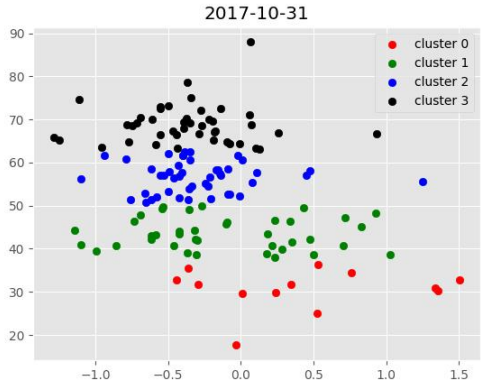

2017-05-31
cluster 0
cluster 1
cluster 2
cluster 3

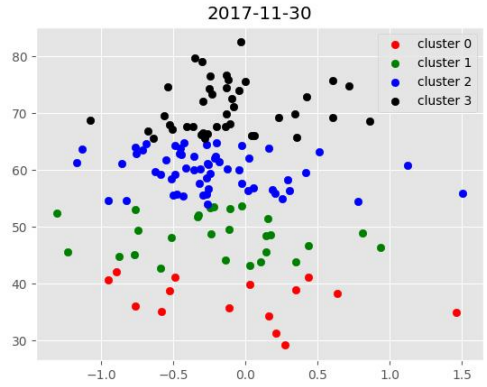

2017-06-30
cluster 0
cluster 1
cluster 2
cluster 3

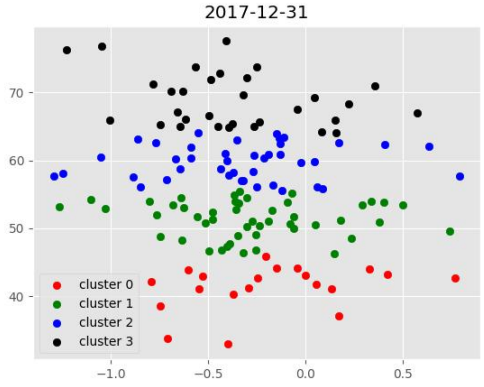

2017-07-31
cluster 0
cluster 1
cluster 2
cluster 3

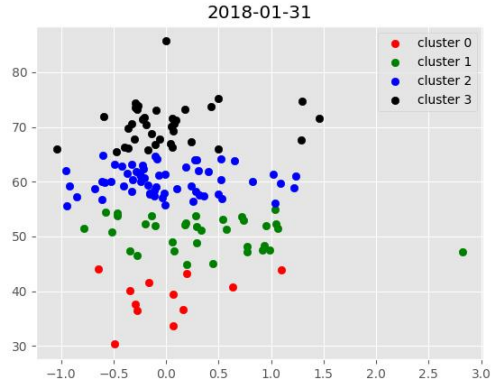

2017-08-31
cluster 0
cluster 1
cluster 2
cluster 3

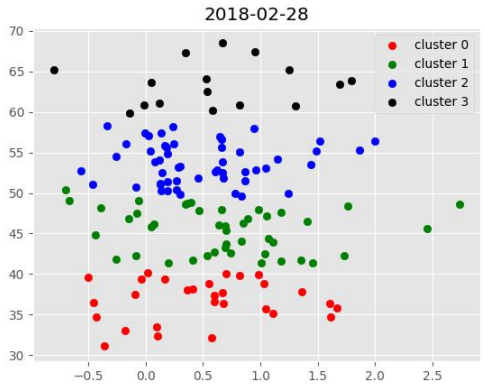

2017-09-30
cluster 0
cluster 1
cluster 2
cluster 3

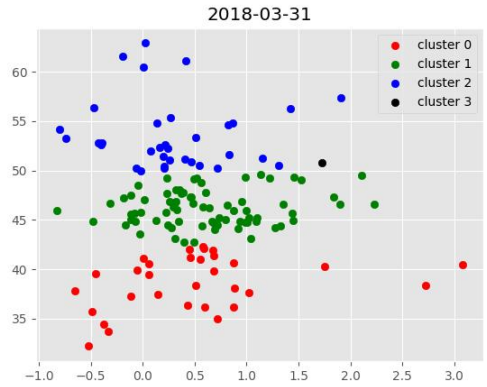

2017-10-31
cluster 0
cluster 1
cluster 2
cluster 3

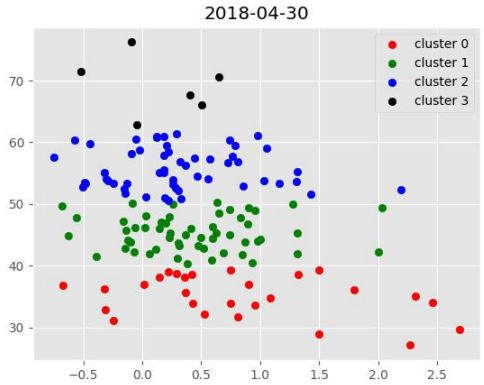

2017-11-30
cluster 0
cluster 1
cluster 2
cluster 3

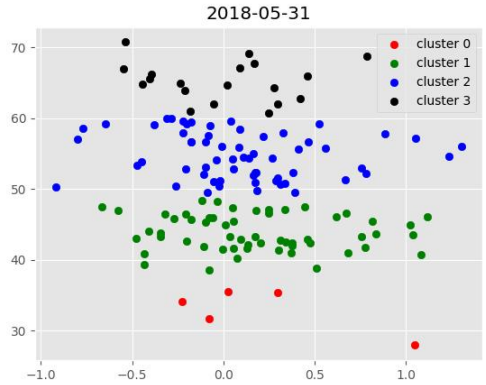

2017-12-31
cluster 0
cluster 1
cluster 2
cluster 3

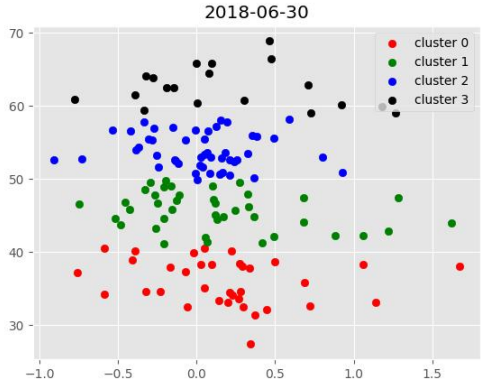

2018-01-31
cluster 0
cluster 1
cluster 2
cluster 3

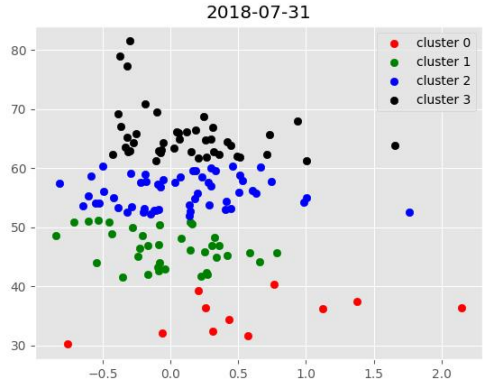

2018-02-28
cluster 0
cluster 1
cluster 2
cluster 3

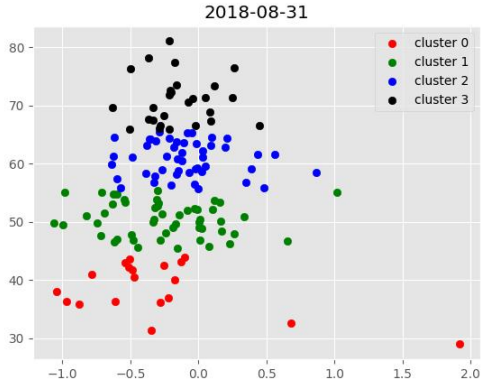

2018-03-31
cluster 0
cluster 1
cluster 2
cluster 3

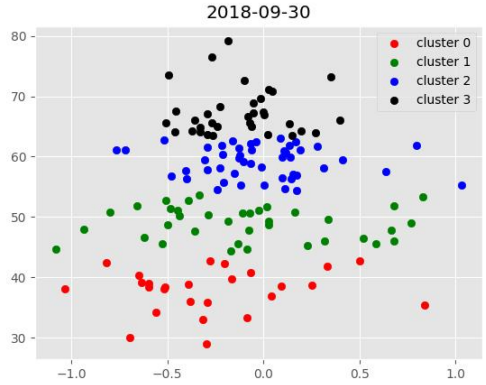

2018-04-30
cluster 0
cluster 1
cluster 2
cluster 3

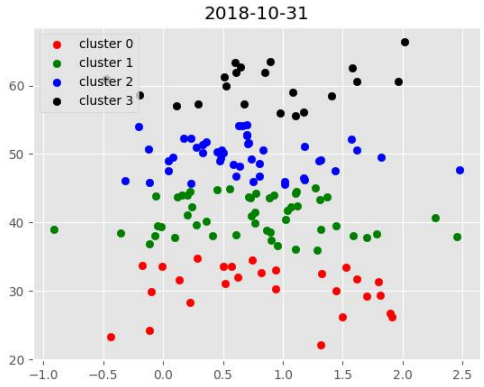

2018-05-31
cluster 0
cluster 1
cluster 2
cluster 3

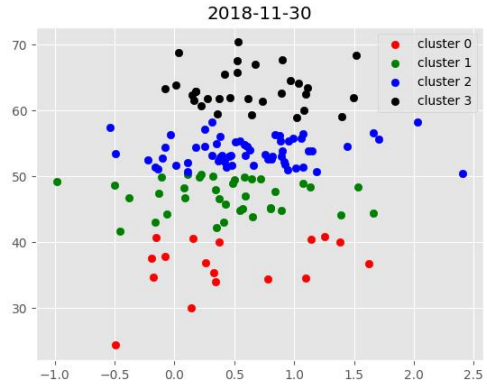

2018-06-30
cluster 0
cluster 1
cluster 2
cluster 3

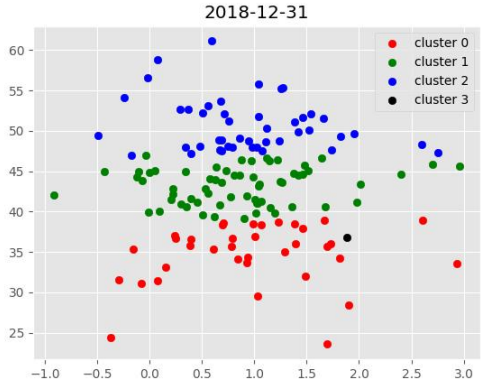

2018-07-31
cluster 0
cluster 1
cluster 2
cluster 3

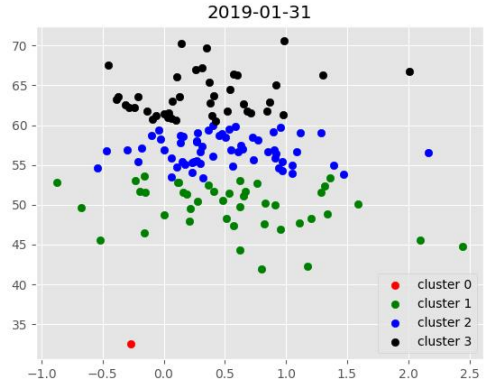

2018-08-31
cluster 0
cluster 1
cluster 2
cluster 3

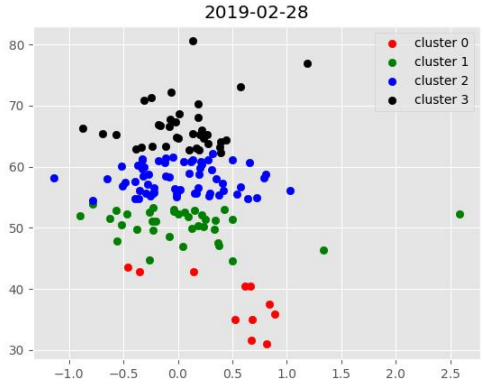

2018-09-30
cluster 0
cluster 1
cluster 2
cluster 3

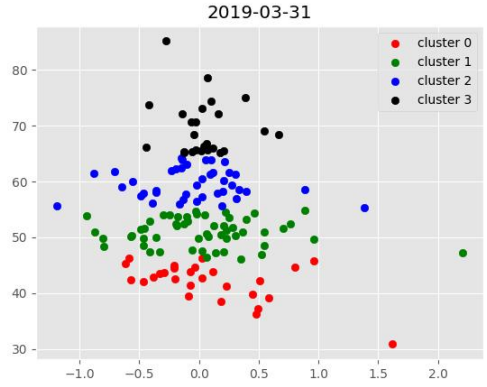

2018-10-31
cluster 0
cluster 1
cluster 2
cluster 3

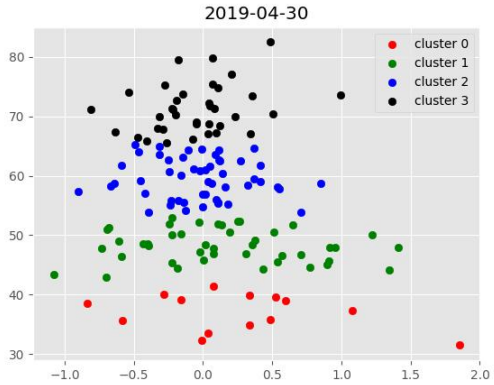

2018-11-30
cluster 0
cluster 1
cluster 2
cluster 3

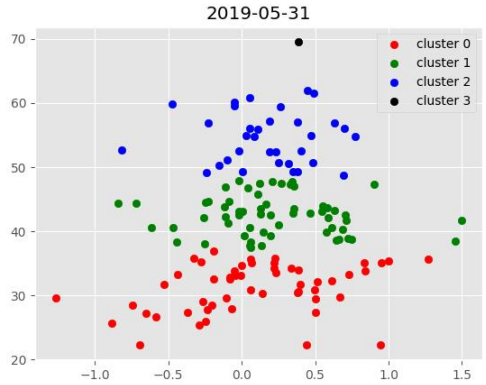

2018-12-31
cluster 0
cluster 1
cluster 2
cluster 3

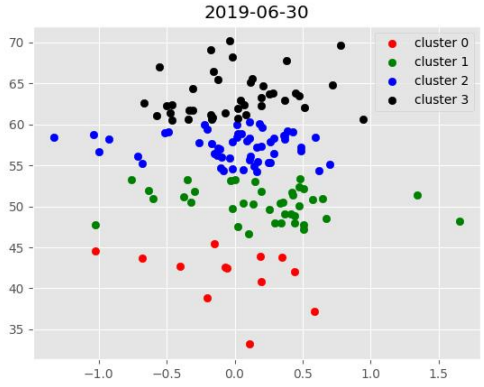

2019-01-31
cluster 0
cluster 1
cluster 2
cluster 3

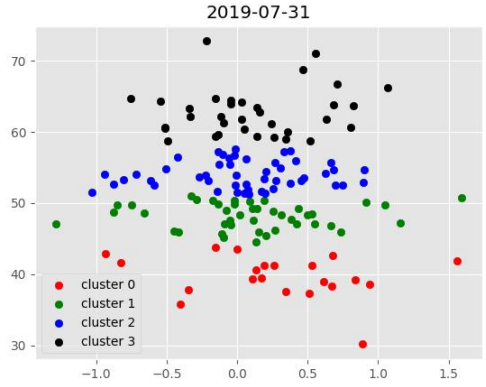

2019-02-28
cluster 0
cluster 1
cluster 2
cluster 3

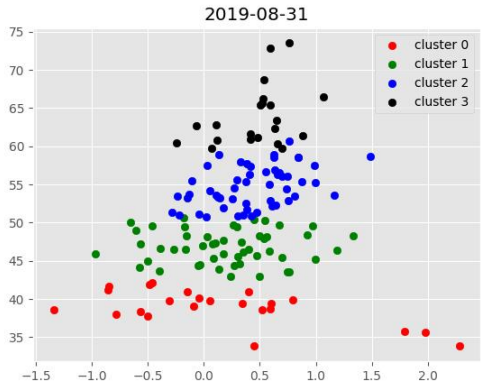

2019-03-31
cluster 0
cluster 1
cluster 2
cluster 3

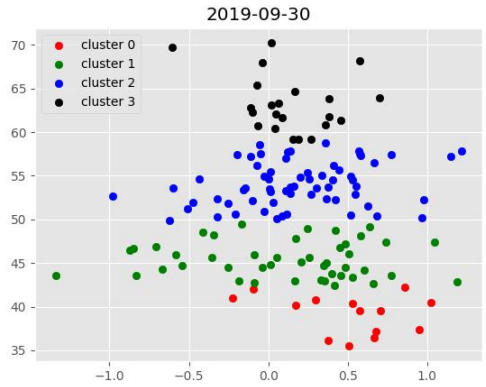

2019-04-30
cluster 0
cluster 1
cluster 2
cluster 3

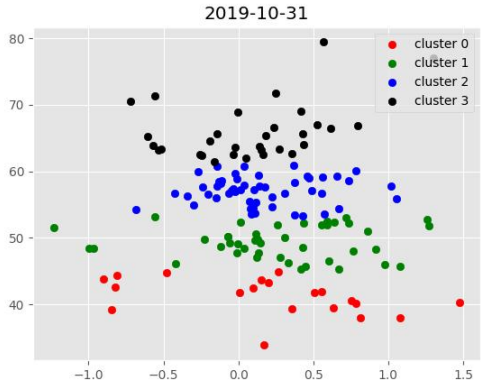

2019-05-31
cluster 0
cluster 1
cluster 2
cluster 3

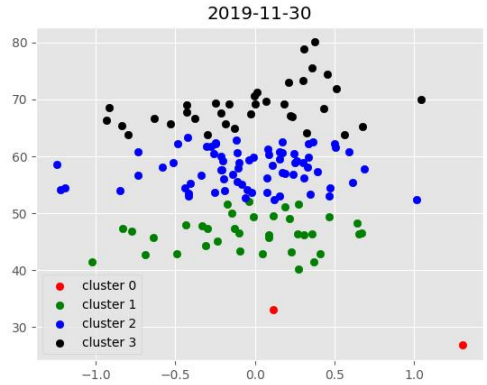

2019-06-30
cluster 0
cluster 1
cluster 2
cluster 3

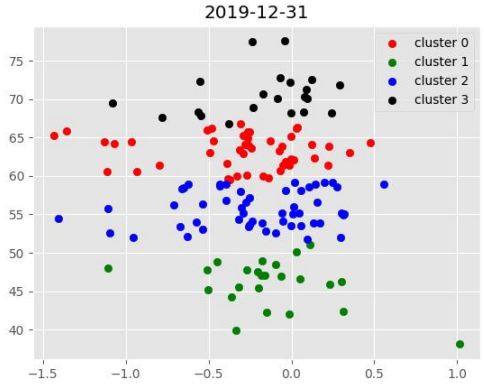

2019-07-31
cluster 0
cluster 1
cluster 2
cluster 3

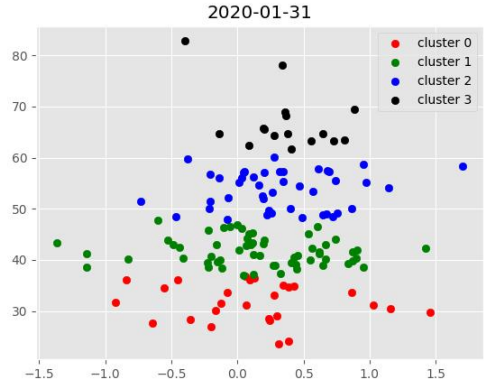

2019-08-31
cluster 0
cluster 1
cluster 2
cluster 3

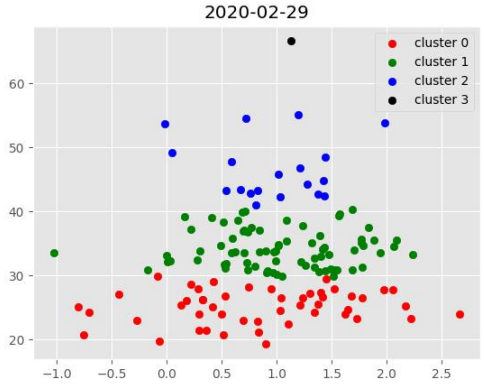

2019-09-30
cluster 0
cluster 1
cluster 2
cluster 3

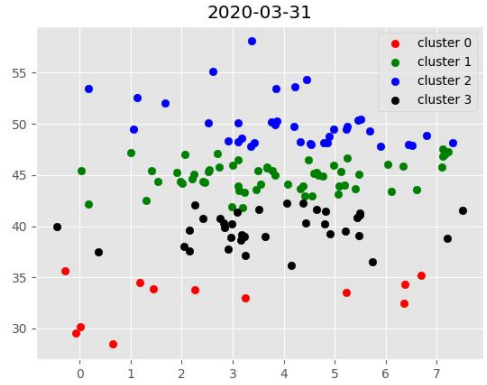

2019-10-31
cluster 0
cluster 1
cluster 2
cluster 3

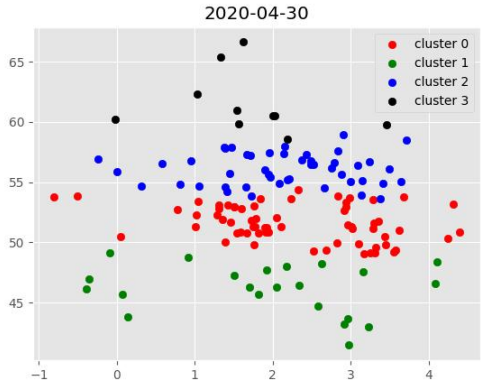

2019-11-30
cluster 0
cluster 1
cluster 2
cluster 3

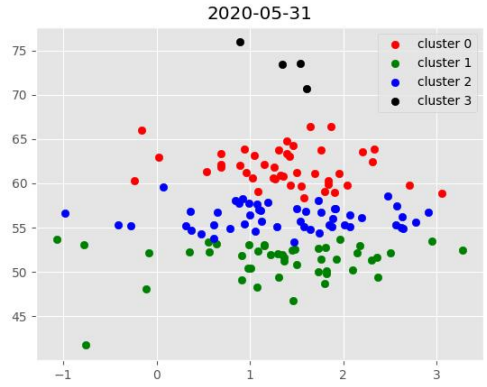

2019-12-31
cluster 0
cluster 1
cluster 2
cluster 3

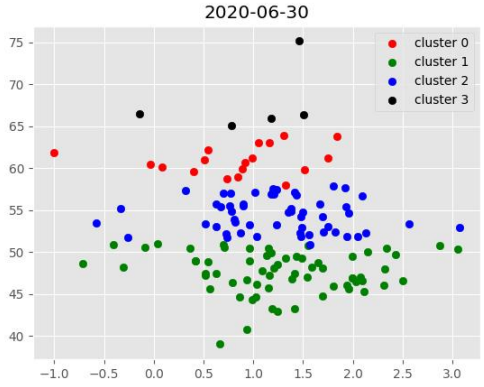

2020-01-31
cluster 0
cluster 1
cluster 2
cluster 3

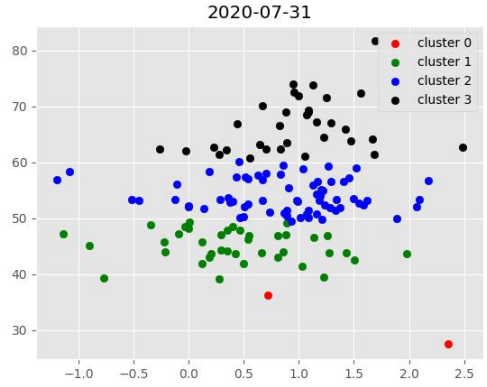

2020-02-29
cluster 0
cluster 1
cluster 2
cluster 3

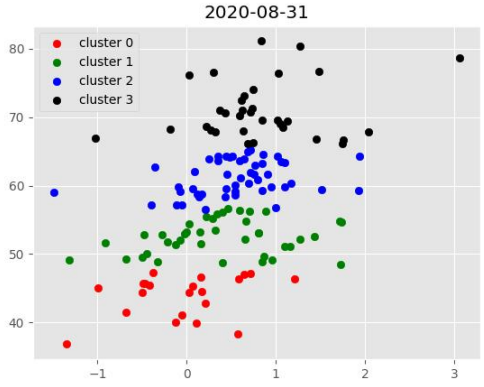

2020-03-31
cluster 0
cluster 1
cluster 2
cluster 3

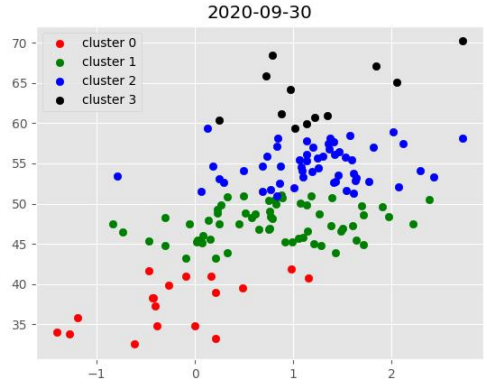

2020-04-30
cluster 0
cluster 1
cluster 2
cluster 3

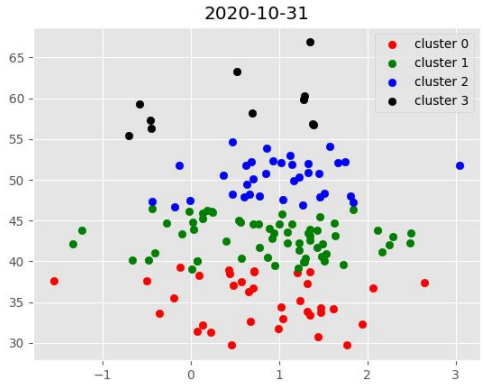

2020-05-31
cluster 0
cluster 1
cluster 2
cluster 3

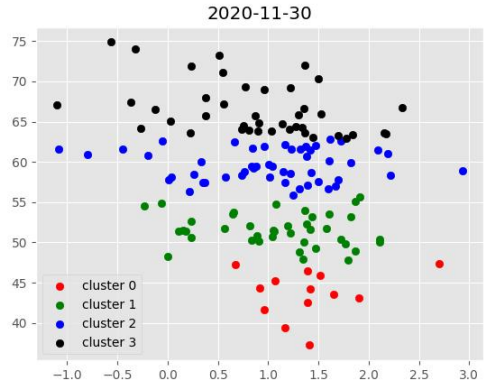

2020-06-30
cluster 0
cluster 1
cluster 2
cluster 3

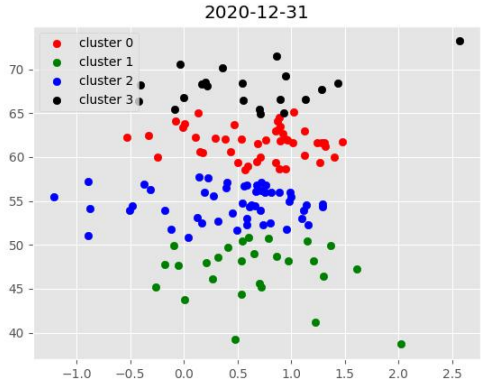

2020-07-31
cluster 0
cluster 1
cluster 2
cluster 3

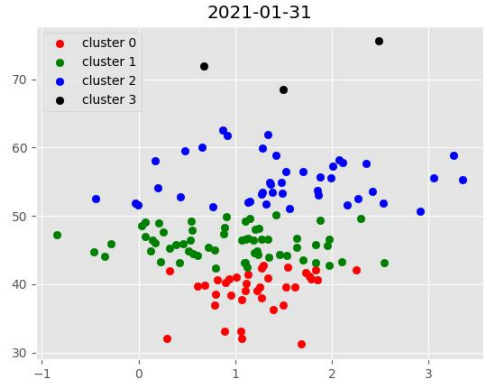

2020-08-31
cluster 0
cluster 1
cluster 2
cluster 3

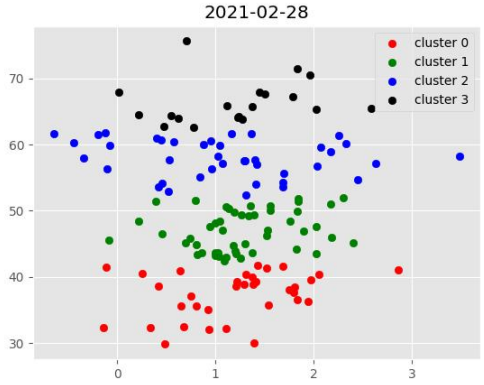

2020-09-30
cluster 0
cluster 1
cluster 2
cluster 3

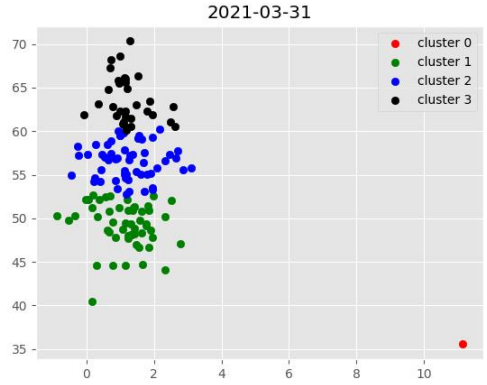

2020-10-31
cluster 0
cluster 1
cluster 2
cluster 3

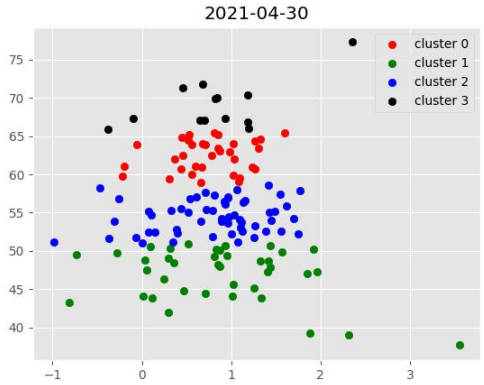

2020-11-30
cluster 0
cluster 1
cluster 2
cluster 3

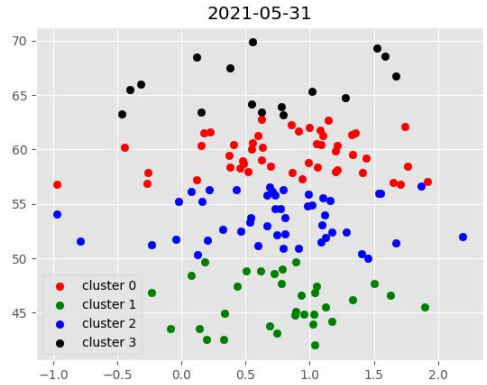

2020-12-31
cluster 0
cluster 1
cluster 2
cluster 3

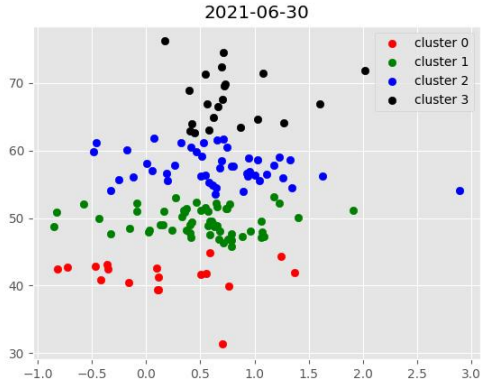

2021-01-31
cluster 0
cluster 1
cluster 2
cluster 3

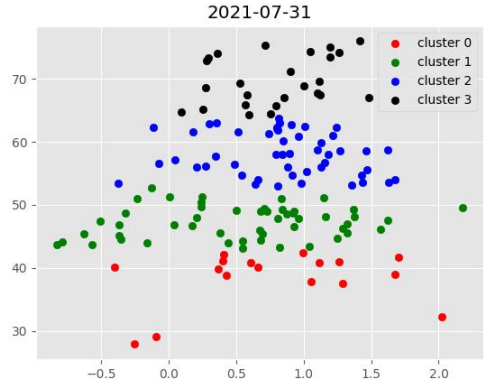

2021-02-28
cluster 0
cluster 1
cluster 2
cluster 3

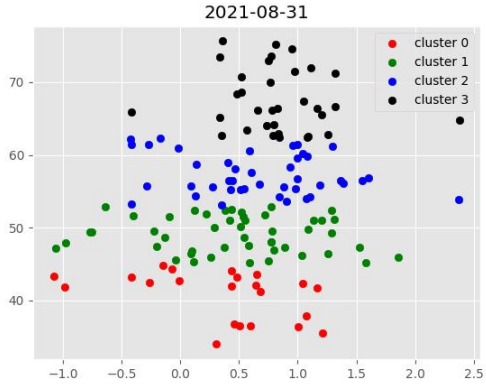

2021-03-31
cluster 0
cluster 1
cluster 2
cluster 3

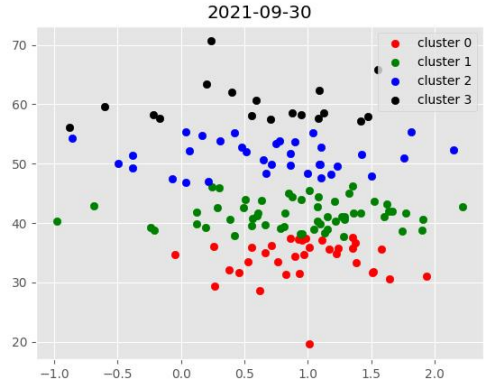

2021-04-30
cluster 0
cluster 1
cluster 2
cluster 3

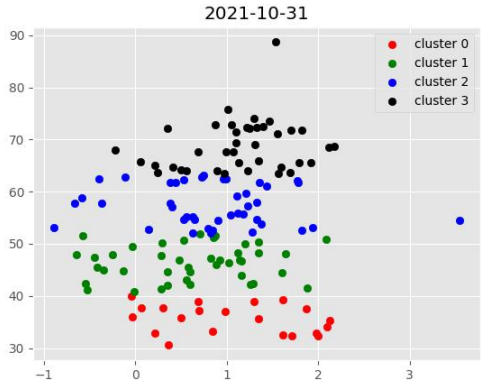

2021-05-31
cluster 0
cluster 1
cluster 2
cluster 3

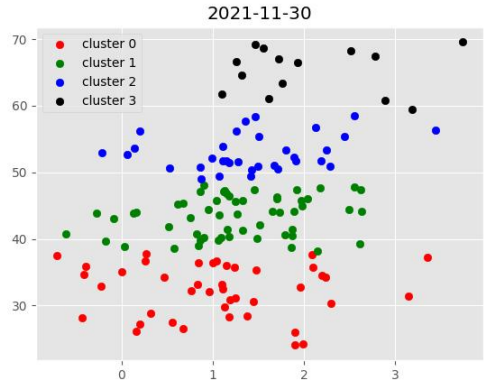

2021-06-30
cluster 0
cluster 1
cluster 2
cluster 3

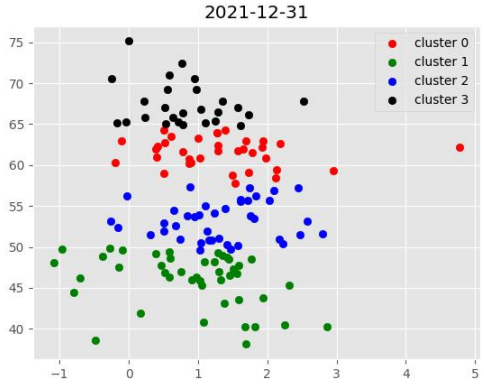

2021-07-31
cluster 0
cluster 1
cluster 2
cluster 3

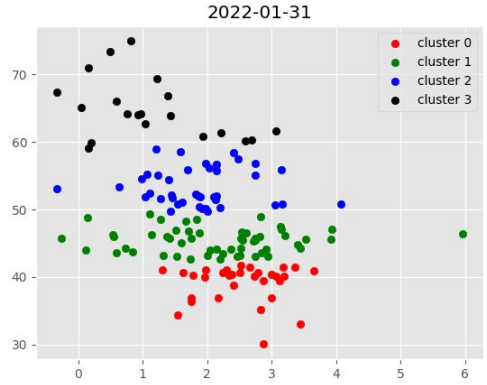

2021-08-31
cluster 0
cluster 1
cluster 2
cluster 3

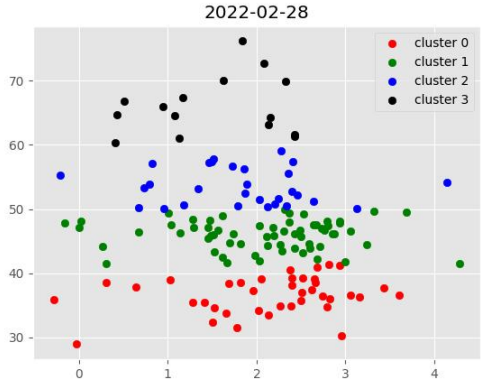

2021-09-30
cluster 0
cluster 1
cluster 2
cluster 3

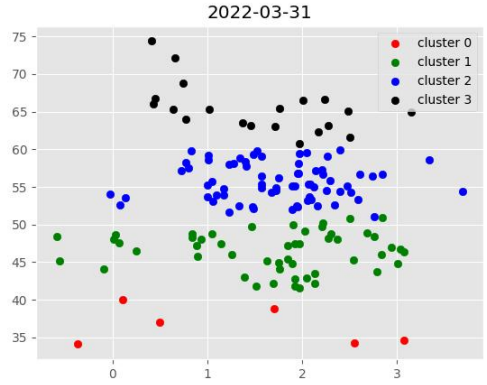

2021-10-31
cluster 0
cluster 1
cluster 2
cluster 3

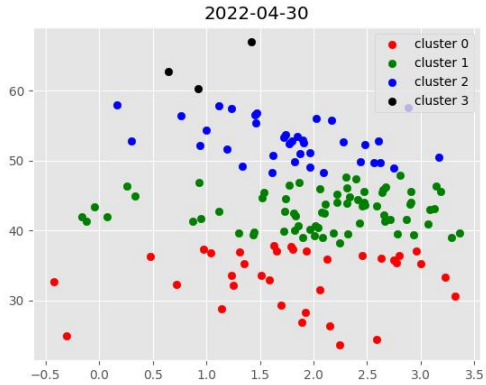

2021-11-30
cluster 0
cluster 1
cluster 2
cluster 3

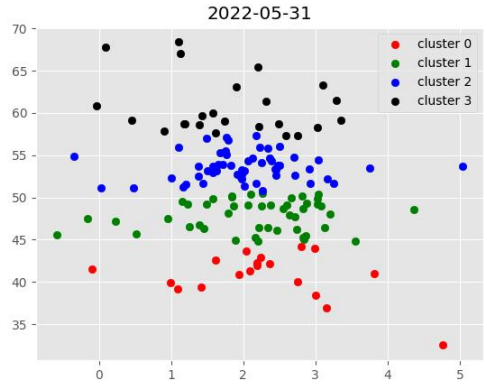

2021-12-31
cluster 0
cluster 1
cluster 2
cluster 3

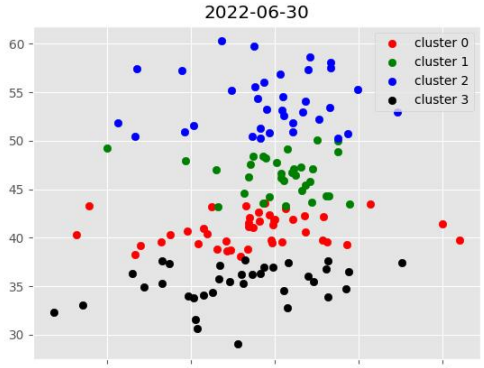

2022-01-31
cluster 0
cluster 1
cluster 2
cluster 3

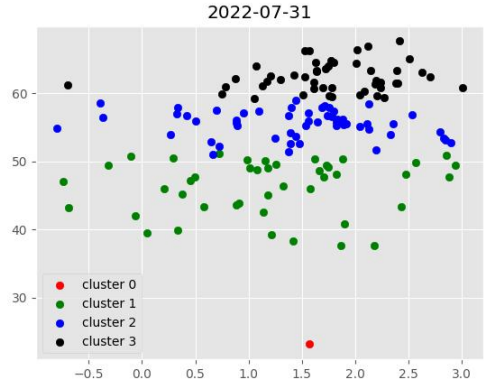

2022-02-28
cluster 0
cluster 1
cluster 2
cluster 3

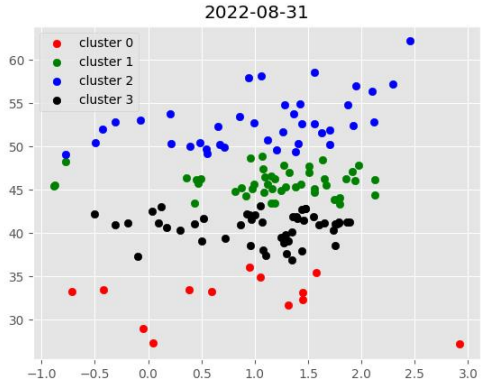

2022-03-31
cluster 0
cluster 1
cluster 2
cluster 3

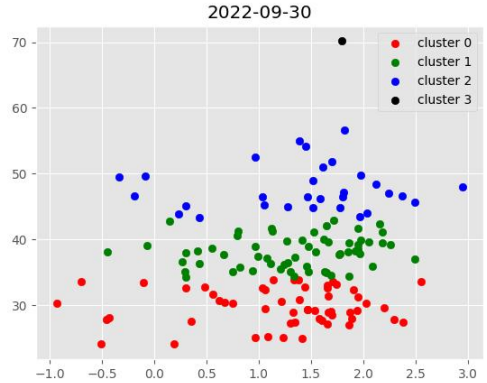

2022-04-30
cluster 0
cluster 1
cluster 2
cluster 3

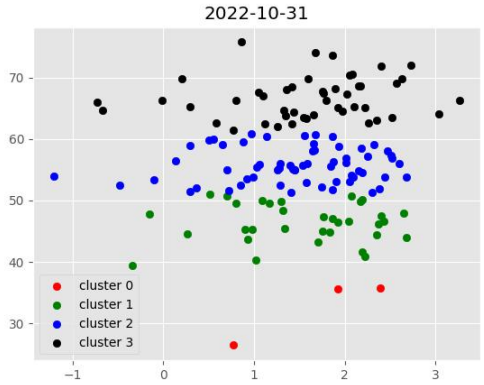

2022-05-31
cluster 0
cluster 1
cluster 2
cluster 3

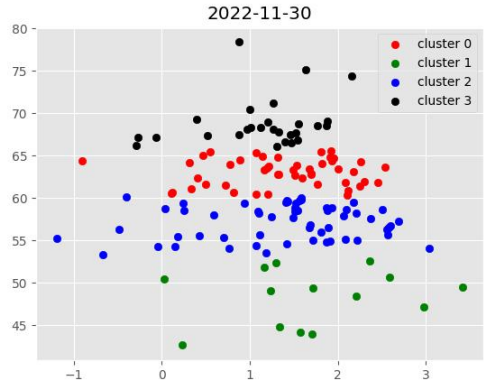

2022-06-30
cluster 0
cluster 1
cluster 2
cluster 3

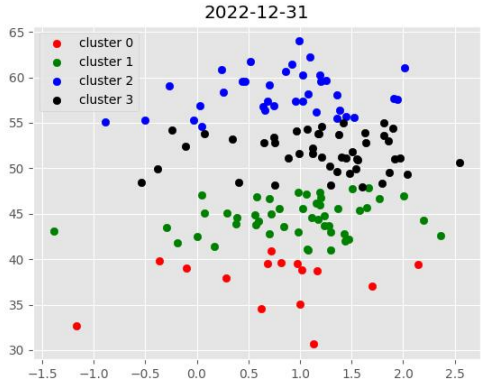

2022-07-31
cluster 0
cluster 1
cluster 2
cluster 3

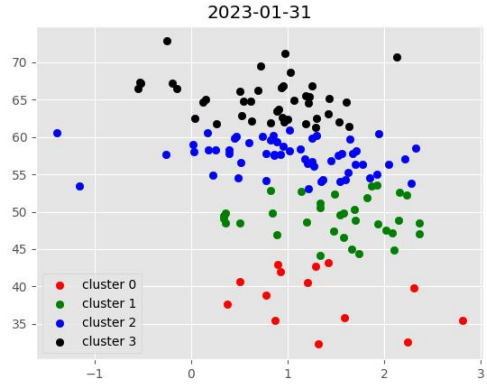

2022-08-31
cluster 0
cluster 1
cluster 2
cluster 3

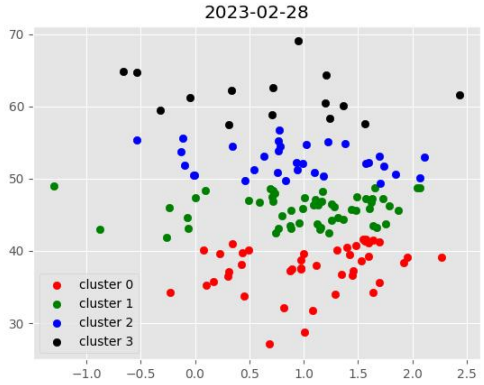

2022-09-30
cluster 0
cluster 1
cluster 2
cluster 3

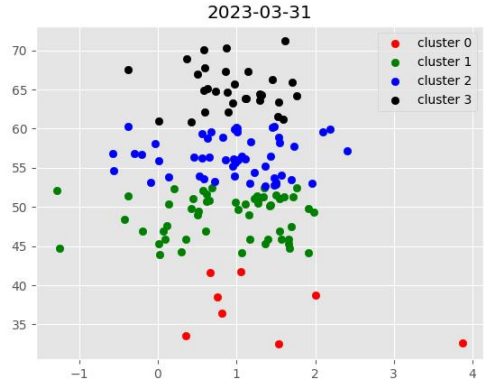

2022-10-31
cluster 0
cluster 1
cluster 2
cluster 3

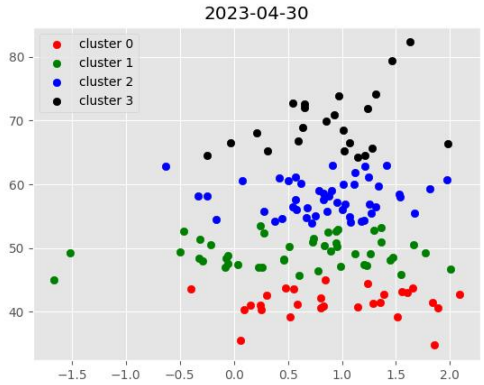

2022-11-30
cluster 0
cluster 1
cluster 2
cluster 3

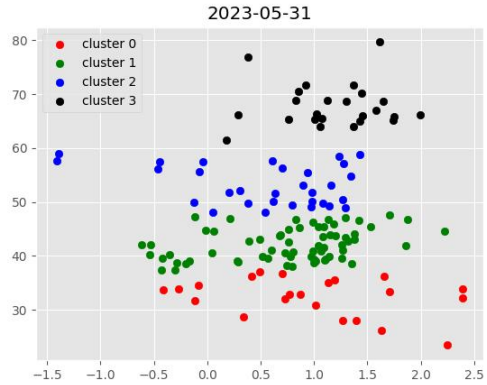

2022-12-31
cluster 0
cluster 1
cluster 2
cluster 3

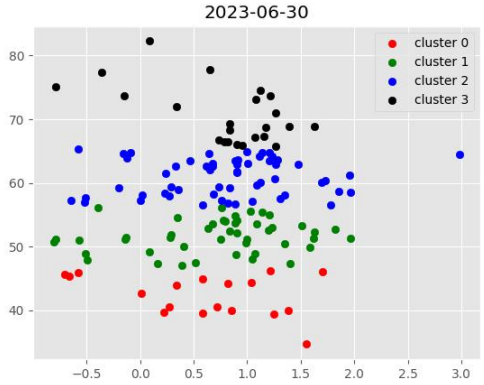

2023-01-31
cluster 0
cluster 1
cluster 2
cluster 3

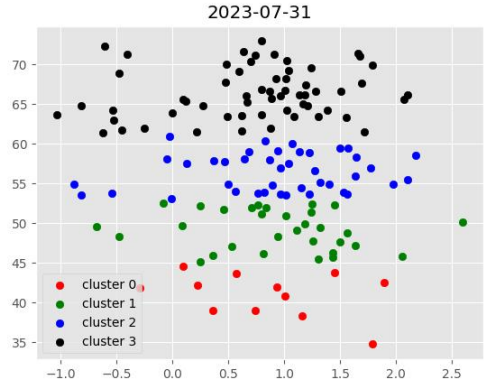

2023-02-28
cluster 0
cluster 1
cluster 2
cluster 3

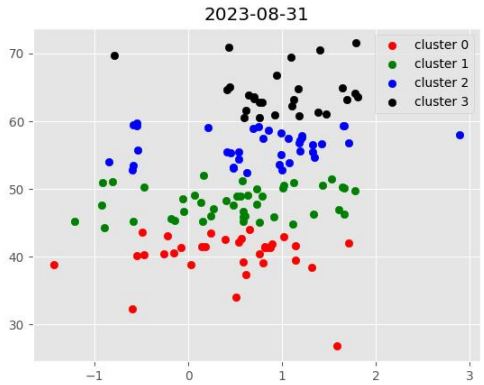

2023-03-31
cluster 0
cluster 1
cluster 2
cluster 3

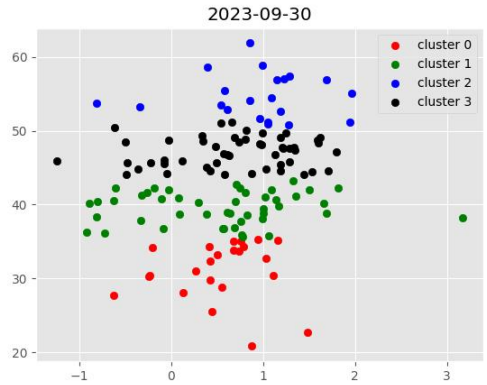

2023-04-30
cluster 0
cluster 1
cluster 2
cluster 3

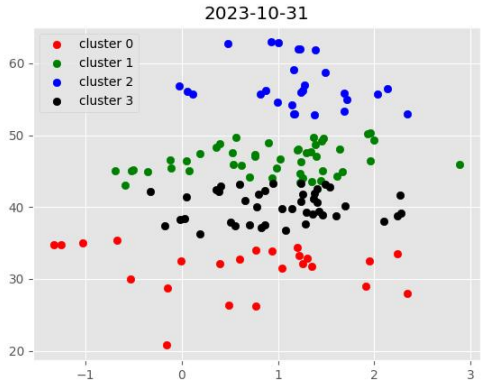

2023-05-31
cluster 0
cluster 1
cluster 2
cluster 3

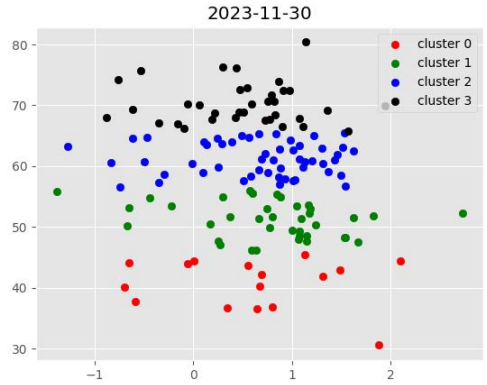

2023-06-30
cluster 0
cluster 1
cluster 2
cluster 3

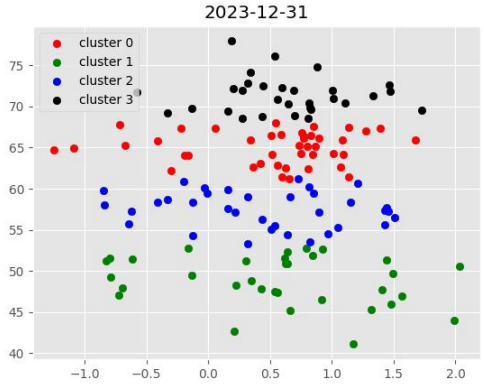

2023-07-31
cluster 0
cluster 1
cluster 2
cluster 3

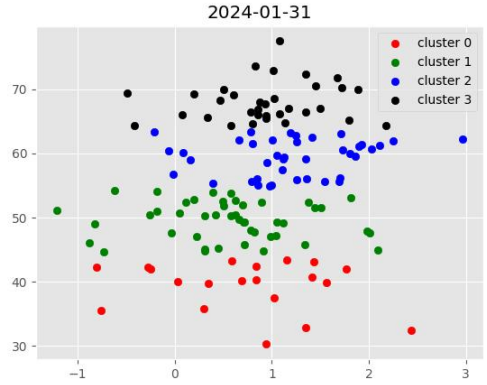

2023-08-31
cluster 0
cluster 1
cluster 2
cluster 3

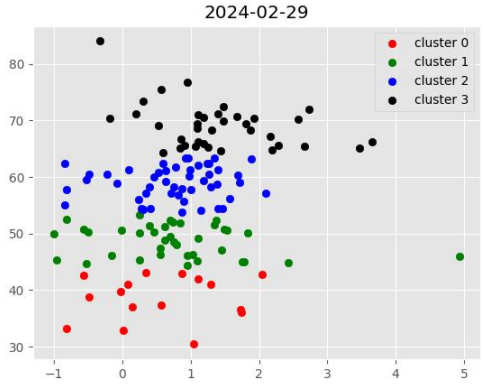

2023-09-30
cluster 0
cluster 1
cluster 2
cluster 3

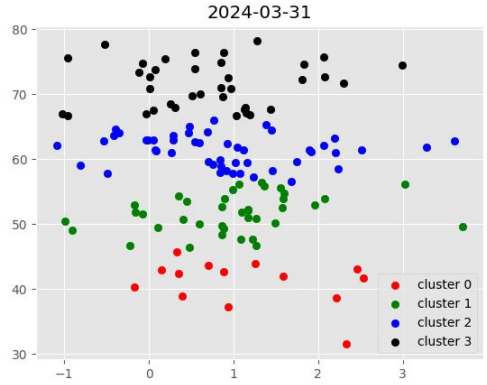

2023-10-31
cluster 0
cluster 1
cluster 2
cluster 3

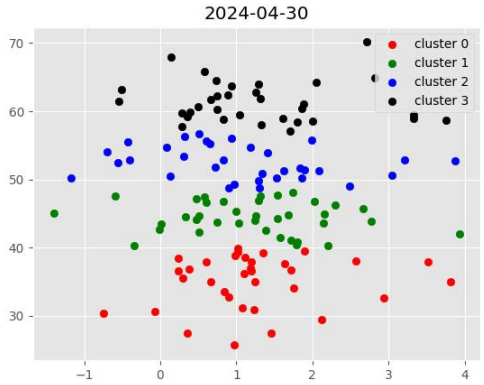

2023-11-30
cluster 0
cluster 1
cluster 2
cluster 3

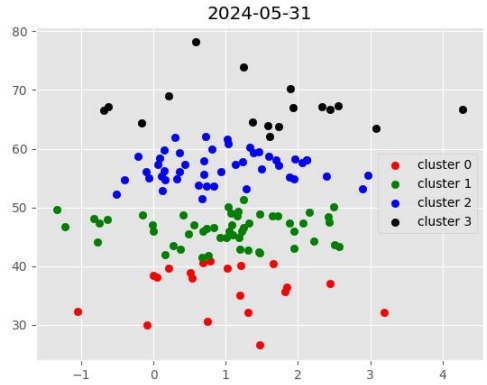

2023-12-31
cluster 0
cluster 1
cluster 2
cluster 3

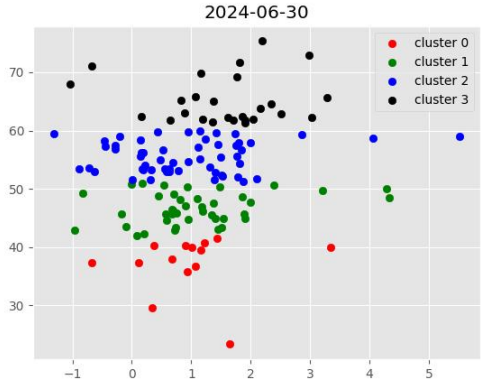

2024-01-31
cluster 0
cluster 1
cluster 2
cluster 3


2024-02-29
cluster 0
cluster 1
cluster 2
cluster 3


2024-03-31
cluster 0
cluster 1
cluster 2
cluster 3


2024-04-30
cluster 0
cluster 1
cluster 2
cluster 3


2024-05-31
cluster 0
cluster 1
cluster 2
cluster 3


2024-06-30
cluster 0
cluster 1
cluster 2
cluster 3

For each month we select assets based on the cluster and form a portfolio based on efficient frontier maximum Sharpe ratio optimisation.

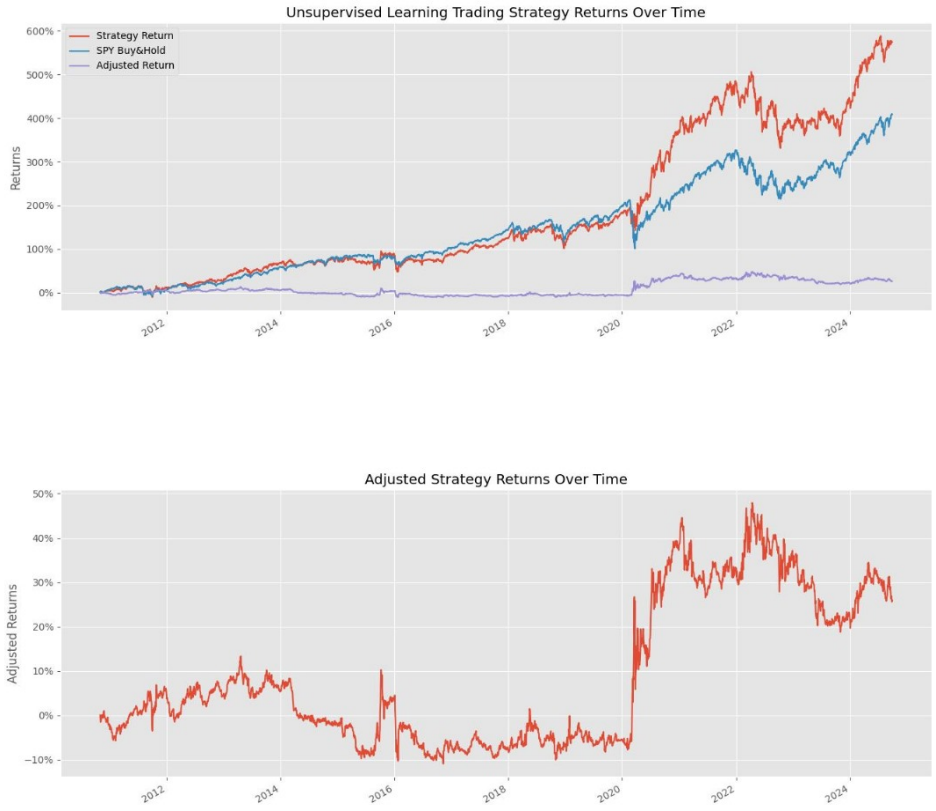
- First we will filter only stocks corresponding to the cluster we choose based on our hypothesis.
- As per our strategy hypothesis we select stocks corresponding to cluster 3.
- We will define a function which optimises portfolio weights using PyPortfolioOpt package and EfficientFrontier optimiser to maximise the Sharpe ratio.
- To optimise the weights of a given portfolio we would need to supply last year's prices to the function.
- Apply single stock weight bounds constraint for diversification (minimum half of equally weighted and maximum 10% of portfolio).

2.4 Computing portfolio returns:

- Calculate daily returns for each stock which could land up in our portfolio.
- Then loop over each month's start, select the stocks for the month and calculate their weights for the next month.
- If the maximum Sharpe ratio optimization fails for a given month, apply equally-weighted weights.
- Calculate daily portfolio returns.

3 Final results

Finally we visualise portfolio returns and compare to S&P 500 returns.



3.1 Limitations and possible future extensions

We do not consider stocks that have been delisted, so there is survivorship bias in our dataset. Yahoo Finance API has been officially discontinued and the current package that we use (yfinance) is an open-source tool that uses Yahoo's publicly available APIs. It is not perfect and returns N/A values for some crucial quantities which is why we had to drop N/A values from the dataframes in our code at various points. This could be mitigated by using a better API with high quality data, but almost all of them are pay-to-use. Also, we haven't split our data so there could be some overfitting that we haven't accounted for.

The model fails to outperform the SPY Buy&Hold over shorter timescales (such as < 5 years). This could be mitigated by extending our current model to use neural networks or other models, and picking the most optimal strategies from the ones given by each model.

4 References

[1] M. F. Dixon, I. Halperin, and P. Bilokon, Machine Learning in Finance. Springer International Publishing, 2020. doi: 10.1007/978-3-030-41068-1.

[2] FreeCodeCamp. Algorithmic Trading – Machine Learning & Quant Strategies Course with Python. (Oct. 26, 2023). Accessed: Sep. 25, 2024. [Online video]. Available: https://www.youtube.com/watch?v=9Y3yaoi9rUQ