Principal Component Analysis in Traditional Stock Investment

Fanyue Meng, Yutong Wu, Haokun Yao

May 9, 2023

Numerical Analysis Final Project

Contents

Abstract

Supposing the market now is already efficient and all the stocks are priced fairly. But in the relatively short time period, the price maybe wrong because investor are not rational and over-reacting to some unexpected news. Then arbitrage will appear.We are able to filter the stocks which are relatively undervalued or relatively overvalued by using PCA. Then we can long undervalued ones and short the overvalued ones. We authenticate our method by our pitching stocks and it works well. High return beat the market.

1 Introduction

Principal Component Analysis (PCA) is a method used to transform high-dimensional data by utilizing the interrelationships between variables in order to create a more manageable and lower-dimensional representation, while still maintaining a significant amount of information. PCA is regarded as one of the most reliable and straightforward techniques for accomplishing this type of dimensionality reduction. In other words, PCA is used to transform a dataset with a large number of variables into a smaller set of uncorrelated variables, called principal components, that capture the most important information in the data.

1.1 Intuition

1.1.1 Dimensionality reduction

Having high-dimensional data can be difficult to analyze and visualize. Moreover, it may cause machine learning models to overfit. PCA minimizes the number of dimensions by transferring the data onto a lower-dimensional space with the greatest possible information preservation.

1.1.2 Data visualization

As we all know, it is difficult to visualize a plot that has greater than 3 dimensions. PCA can lower the number of dimensions, helping in identifying patterns, trends, and outliers in the data.

1.1.3 Noise reduction

PCA helps eliminate noise in the data by identifying and preserving the most significant features that explain the majority of the variance in the data. This results in a more accurate representation of the underlying structure and relationships.

1.1.4 Computational efficiency

Reducing the number of dimensions can significantly speed up the computation time for various algorithms, especially those that have high computational complexity.

1.1.5 Data compression

Similar to truncated SVD, PCA can be used to compress data by retaining only the most important components, which reduces storage and computation requirements without losing too much information.

2 General Steps

1. Normalized the data: set the mean equal to 0 and the standard deviation to 1 for each feature.

2. Calculate the covariance matrix: calculate the covariance matrix for the normalized data which captures the relationship between the features in the data set.

3. Singular Value decomposition: Compute the eigenvectors and corresponding eigenvalues of the covariance matrix. Analyze which factor will explain the most variance. Eigenvectors represent the directions of the principal components, while eigenvalues represent their magnitude or the amount of variance they explain.

4. Sort the eigenvectors by eigenvalue: Arrange the eigenvectors in descending order according to their corresponding eigenvalues. The eigenvector with the highest eigenvalue is the first principal component, the one with the second-highest eigenvalue is the second principal component, and so on.

5. Select the number of principal components: Project the original data onto the principal components by multiplying X by V,(we do need to do the reduced SVD on covariance matrix, instead we can do the reduced SVD directly on X) which represents the original data in the principal component space.

3 Methodology

3.1 Math Part

3.1.1 Normalization

Features in a dataset often have different units and scales. Without normalization, features with larger scales or units could dominate the analysis, leading to biased results. Normalization standardizes the features, bringing them to the same scale, so that each feature contributes equally.
By setting the mean to 0 and standard deviation to 1 for each feature, you ensure that PCA captures the underlying structure of the data without being influenced by the original scales of the features.

To calculate the mean of a set of values for a given feature, you need to sum up all the values for that feature and divide the result by the number of observations (data points) in your dataset.

Let's consider a $m * n$ matrix X

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

Each column of matrix X represents a feature so that there are n features for matrix X. There are m rows, which means that the number of observations is m.

The mean $\overline{x}$ of a set of values for a given feature can be calculated by

$$\overline{x} = \frac{1}{m}(x_1 + x_2 + \cdots + x_i + \cdots + x_m)$$

To calculate the standard deviation of a set of values for a given feature, you first need to compute the variance and then take the square root of the variance. The formula for the sample variance is as follows:

$$var(x) = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \overline{x})^2$$

m is the number of observations (data points) in your dataset
$\sum_{i=1}^{m}(x_i - \overline{x})^2$ is the sum of the squared differences between each value x_i and the mean of the feature, $\overline{x}$.

In PCA, we use sample variance because we typically work with a sample of data from a larger population. The data points we have at hand are assumed to be a representative subset of the entire population. Since we do not have access to the whole population, we must estimate the population variance using the sample data we have.
Sample variance is used as an unbiased estimator of the population variance. When we calculate the variance using the formula with the denominator $m - 1$ instead of m, it corrects for the bias that arises from using the sample mean as an estimate of the population mean.

Once you have the sample variance, you can find the standard deviation s by taking the square root:

$$s = \sqrt{var(x)}$$

Finally, we can do normalization for each data, which is also known as z-score normalization.

$$\tilde{X} = \frac{(X - \overline{x})}{s}$$

$\overline{x}$ is mean of each feature and s is standard deviation of each feature.

Normalization affects the mean and standard deviation of the data but does not change the relative positions of the data points in the distribution. The overall shape of the distribution remains the same, but it is now centered around 0 with a standard deviation of 1.

Consider the following graph,

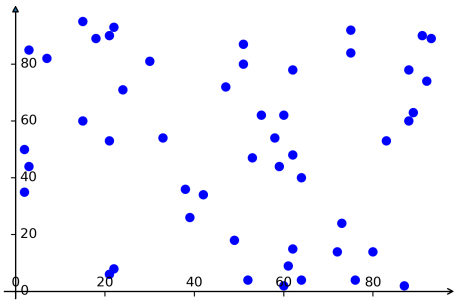

Figure 1: Original data

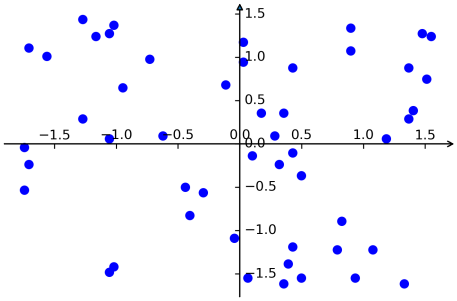

Figure 2: Normalized data

We can clearly see that the distribution of data remains the same.

If you don't perform normalization before applying PCA, the results may be biased towards the features with larger scales. PCA is sensitive to the scale of the input features because it aims to maximize the variance captured by the principal components. Features with larger scales will naturally have higher variance, causing the principal components to be aligned with these features, even if they are not necessarily the most informative ones.

3.1.2 Covariance

In PCA, computing the covariance matrix is an essential step as it captures the linear relationships between the features in the dataset. The covariance matrix represents the degree to which features vary together. It helps to identify the directions of maximum variance in the data, which are then used to form the principal components.

Now, we are ready to calculate the covariance matrix using normalized data. We compute the covariance matrix to quantify the relationships between the different features in the dataset, capturing both the variance of individual features and the correlations between pairs of features.

The covariance between two feature, x and y, can be expressed as,

$$cov(x, y) = \frac{1}{m-1} \sum_{i=1}^{m} (x_i - \bar{x})(y_i - \bar{y})$$

x_i and y_i are the individual data points for features x and y, respectively. $\bar{x}$ and $\bar{y}$ are the means of features x and y, respectively. m is the number of data points. Since we are calculating sample covariance, we need to us the formula with the denominator $m-1$ instead of m as explained above.

The variance of a feature x, denoted as $var(x)$ or s^2, can also be represented as the covariance between x and itself,

$$\begin{aligned} cov(x, x) &= \frac{1}{m-1} \sum_{i=1}^{m} (x_i - \bar{x})(x_i - \bar{x}) \\ &= \frac{1}{m-1} \sum_{i=1}^{m} (x_i - \bar{x})^2 \\ &= var(x) \end{aligned}$$

If we use the normalized data $\tilde{X}$, which set mean equal to 0, the mean of each feature will become 0.

Thus, we can simplify it as

$$cov(x, y) = \frac{1}{m-1} \sum_{i=1}^{m} (x_i)(y_i)$$

x_i and y_i represent entries of different feature from $\tilde{X}$.

3.1.3 Covariance Matrix

Given a dataset with n features, the covariance matrix will be an $n * n$ matrix. Each entries (i, j) in the covariance matrix represents the covariance between feature i and feature j. The diagonal elements of the covariance matrix correspond to the variances of the individual features.

Now, let's look at the covariance matrix, a square matrix by definition.

$$C = \begin{bmatrix} cov(x_1, x_1) & \cdots & cov\,(x_1, x_n) \\ \vdots & \cdot & \vdots \\ \vdots & & \vdots \\ cov\,(x_n, x_1) & \cdots & \mathrm{cov}\,(x_n, x_n) \end{bmatrix}$$

Since $cov(x_j, x_j)$ is equivalent to $var(x_j)$, we also can express the covariance matrix as

$$C = \begin{bmatrix} var(x_1) & \cdots & cov\,(x_1, x_n) \\ \vdots & \cdot & \vdots \\ \vdots & & \vdots \\ cov\,(x_n, x_1) & \cdots & \mathrm{var}\,(x_n) \end{bmatrix}$$

The diagonal entries of the covariance matrix are the variances.

It is also important to notice that the covariance matrix is a symmetric matrix since $cov(x_i, x_j) = cov(x_j, x_i)$, which means that $C = C^T$.

Each entry (i, j) in the covariance matrix can be represented as

$$cov(x_i, x_j) = \frac{1}{m-1} \sum_{k=1}^{m} (x_{ik})\,(x_{jk})$$

if we use data from $\tilde{X}$

x_i and x_j are different features and m is number of observation.

Thus, if we are using the normalized data $\tilde{X}$, the covariance matrix can be expressed as,

$$C = \frac{1}{m-1} \tilde{X}^T \tilde{X}$$

3.1.4 Singular value decomposition

In the context of PCA, SVD aims to find a projection that maximizes the variance of the projected data. Each eigenvector of the covariance matrix represents the direction of the principal component, and its corresponding eigenvalue indicates the amount of variance captured by that component. We want to find the largest magnitude of the eigenvalue, which will tell us the most variance. By sorting the eigenvectors in decreasing order of their corresponding eigenvalues, the top two eigenvalues and eigenvectors determine the most important directions of variation in the data. These eigenvectors serve as the basis for the new lower-dimensional space which captures the most variance in the data.

3.1.5 SVD of Covariance Matrix

We know that Covariance Matrix can be expressed as:

$$C = \frac{1}{m-1}\tilde{X}^T\tilde{X}$$

If we do SVD on Covariance Matrix,

$$C = \frac{\tilde{X}^T\tilde{X}}{m-1}$$

$$= \frac{1}{m-1}\left(U\Sigma V^T\right)^T\left(U\Sigma V^{\mathrm{T}}\right)$$

$$= \frac{1}{m-1}V\Sigma^{\mathrm{T}}U^{\mathrm{T}}U\Sigma V^{\mathrm{T}}$$

$$= \frac{1}{m-1}V\Sigma^{\mathrm{T}}\Sigma V^{\mathrm{T}}$$

$$= \frac{1}{m-1}V\Sigma^{\mathrm{T}}\Sigma V^{\mathrm{T}}$$

$$= \frac{1}{m-1}V\Sigma^2V^{\mathrm{T}}$$

Notice that since U is an orthogonal matrix which means that

$$U^{\mathrm{T}} = U^{-1}$$

So that,

$$U^{\mathrm{T}}U = U^{-1}U = I$$

Now, we can express the covariance matrix as

$$C = \begin{bmatrix} | & & | \\ [V]_{:,1} & \cdots & [V]_{:,n} \\ | & & | \end{bmatrix} \begin{bmatrix} \frac{\sigma_1^2}{m-1} & & \\ & \ddots & \\ & & \frac{\sigma_n^2}{m-1} \end{bmatrix} \begin{bmatrix} -[V]_{:,1}^T- \\ \vdots \\ -[V]_{:,n}^T- \end{bmatrix}$$

3.1.6 Principal Component Analysis

Since each eigenvector of the covariance matrix represents the direction of the principal component. If we multiply our data matrix X by V, then we get a new matrix with transformed data. The transformed data is called principal component scores by projecting the original data onto the principal components which represents the original centered data in the principal component space.

If we want to transform the data into 2 dimension space, we should extract v_1 and v_2 which capture the most variance where v_1 and v_2 contains the coefficients of PC1 and PC2.

$$XV = X \begin{bmatrix} | & & | \\ [V]_{:,1} & \cdots & [V]_{:,n} \\ | & & | \end{bmatrix}$$

We also can expressed XV as following,

$$X = U\Sigma V^{\mathrm{T}}$$

since V is orthogonal matrix, $V^T = V^{-1}$

$$X = U\Sigma V^{-1}$$

$$XV = U\Sigma V^{-1}V$$

$$XV = U\Sigma$$

Thus, we only need to find singular decomposition for $\tilde{X}$ instead of finding the covariance matrix of $\tilde{X}$ to get the principal component. We also can get the eigenvalue of the covariance matrix by using the eigenvalue of $\tilde{X}$. Eigenvalue of C is $\frac{\sigma^2}{m-1}$, σ is eigenvalue of $\tilde{X}$.

3.1.7 Toy Example

In PCA, the top k eigenvectors (principal components) capture the most variance in the data, which means they contain the most information about the original dataset.

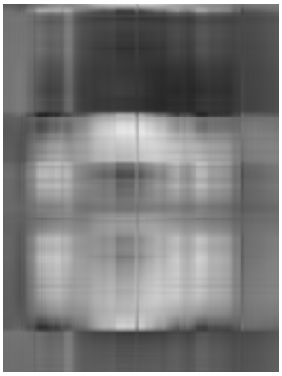

Figure 3: reconstructed image 2 components

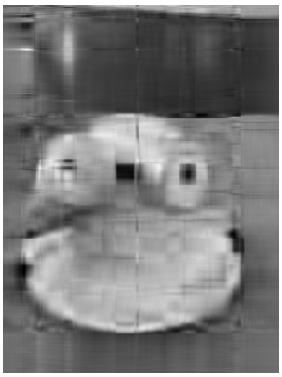

Figure 4: reconstructed image 10 components

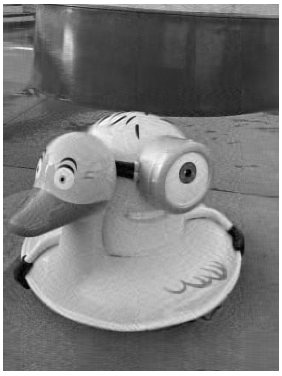

Figure 5: reconstructed image 50 components

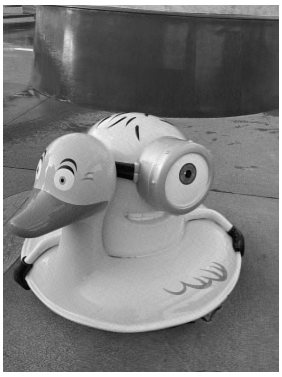

Figure 6: reconstructed image 100 components

The dimensions of the original image: 4032 x 3024. Since the top k eigen-

vector contains the most information, we can compress data using less space. As we can see, the reconstructed image with only 50 components already looks similar to the original image.

3.2 Finance Part

3.2.1 Definition and Properties of the Finance Terms

Assumption:
We assume that the stock should be priced according to its intrinsic value. Investors are risk neutral, which means they are willing to take risk at any level. But due to some short-term unexpected reason, the price of the stock will be undervalued or overvalued because people handle the information inappropriately, causing the price to go off the right track. In this project, we assume that this misleading price phenomenon continues, and we want to find out the stock that the investor can profit by long or short. This action will also help the stock to return to its right price. It is beneficial for both rational investors and the markets.

For this PCA case study, we will find 5 PCA factors. These five factors are key factors in studying the intrinsic value of a company stock:
Definitions and Properties of them are shown below:

Price to Earning: The price-to-earnings ratio indicates the dollar amount an investor can expect to invest in a company in order to receive 1 dollar of that company's earnings. **Low Price Earnings Ratio businesses are frequently referred to as value stocks.** Their stock prices trade below their fundamentals, which indicates that they are **undervalued**. Investors will purchase the stock as a result of this mispricing in order to take advantage of the fantastic deal before the market corrects it.

$$P/E = \frac{Share\ price}{Earnings\ per\ share} \tag{1}$$

Earning Per Share: Earnings per share (EPS) is calculated as a company's profit divided by the outstanding shares of its common stock. The resulting number serves as an indicator of a company's profitability. **The higher this term, the company earning ability is much stronger. It means the company is valuable.**

$$EPS = \frac{net\ income\ -dividend\ payments}{weighted\ average\ shares\ outstanding} \tag{2}$$

Debt to Equity: Debt-to-equity (D/E) ratio is used to evaluate a company's financial leverage and is calculated by dividing a company's total liabilities by its shareholder equity. The D/E ratio is an important metric in corporate finance. It is a measure of the degree to which a company is financing its operations with debt rather than its own resources. **We want this term smaller. The smaller the term, the better the company uses its own**

resources and lowers the risk of the company since they use little leverage. Leverage has the effect of magnification, it will not only magnify the profit, but it will also magnify the loss. Higher leverage normally means the company is less valuable.

$$Debt\ to\ Equity = \frac{Total\ Liabilities}{Total\ Shareholders'\ Equity} \tag{3}$$

Price to Book: The price-to-book value (P/B) ratio measures the excess of a company's book value of equity over the market value of its shares, or share price. The worth of a company's assets(book value) as shown on the balance sheet is the book value of equity. **The lower the P/B, the better the stock, the potential rise of price in the future there will be.** Investors may be alerted that a stock is undervalued by a P/B ratio with lower values, particularly those below one.

$$P/B\ ratio = \frac{market\ price\ per\ share}{book\ value\ per\ share} \tag{4}$$

Price to 52W range: Price to 52-week range is a financial ratio that compares a stock's current price to its highest and lowest prices over the past 52 weeks. The price to 52-week range ratio is used as an indicator of a stock's relative value and potential for future growth. **A high ratio may indicate that the stock is overvalued, while a low ratio may indicate that the stock is undervalued.**

$$Price\ to\ 52W\ range = \frac{Current\ price}{52W\ high\ -\ 52W\ low} \tag{5}$$

3.2.2 Reference Point

As we have mentioned in last section, the lower the Price to Earning, Debt to Equity, Price to Book, Price to 52W, and the higher the Earning Per Share, the better a stock it will be. After we normalize the data, the result will show from 0 to 1. For the lower the better data, as they are closer to 0, the better they will be. As Earning Per Share is closer to 1, the better the stock it will be. First we subtract all the lower better data with 1, so as **they are close to 1**, the better they will be.
In other words, we set five reference points equal to 1. After PCA of the modified data, if they are close to reference point, it means they are closer to the ideal situation, and we should consider them more.

4 Case Study

In this part, we are going to find out the stock that is possibly undervalued or in other words, worth buying according to five factors we mentioned in the methodolgy part.

4.1 Dig out the undervalued stock

In the traditional investment process, there are five factors that we value the most: Price to Earnings, Earnings per Share, Debt to Equity, Price to Book, and Price to 52W Range. It is worth noting that a change in the choice of factors could lead to a different result. So if the reader wants to consider other factors, he can use the code in the appendix.

First, we pull the data of the stock, as shown in the following:

Out[6]:

Stock	Price to Earnings	Earnings Per Share	Dept to Equity	Price to Book	Price to 52W Range
AAPL	27.872232	5.87	195.868	45.688354	51.9800
ABCL	13.450000	0.50	6.670	1.564317	9.5500
ABNB	41.467155	2.74	42.104	12.893781	81.3300
ADBE	35.690945	10.16	29.044	11.716316	176.4200
AMD	102.244060	0.84	5.399	2.528707	55.0000
APPS	24.346940	0.49	71.274	1.953176	25.5900
ASML	31.906970	19.08	40.596	1363.380900	335.4400
AVGO	20.781904	29.62	169.009	11.012004	233.4300
AZPN	62.482430	3.70	2.795	1.138910	114.6600

Figure 7: Part of the Raw Data.

We then need to normalize all the data. The reason we are doing that is we want to put the data on the same scale and in this way, we are able to avoid the larger value dominating the result of the PCA. Here is the formula for the PCA fitting into range 0 to 1:

$$x_{normaliztion} = \frac{x - mean}{std} \tag{6}$$

$$x_{scaled} = \frac{x - \min(column)}{\max(column) - \min(column)} \tag{7}$$

$$X_{scaled} = x_{normaliztion} * (1 - 0) + 0 \tag{8}$$

After doing this, we are able to get the normalized data. The following figure shows the normalized data:

The reader is able to see a reference point on the last row, which is no shown in the following picture. It is a point that will be used in the further data procession, and it is the ideal point we want to achieve, which will be 1 by definition.

Since those factors show a property that the lower, the better. In other words, if the price to earning is smaller, earning per share is smaller, debt to equity is smaller, etc. The investor will be more willing to invest in stocks with these properties. As the normalized data ranges from 0 to 1, so if the data is closer to

	Stock	Comp1	Comp2	Comp3	Comp4	Comp5
0	AAPL	0.581231	0.258899	0.114285	0.642321	0.318084
1	ABCL	0.163264	0.448385	0.062049	0.586840	0.466807
2	ABNB	0.266141	0.440819	0.146069	0.590901	0.495279
3	ADBE	0.325669	0.605835	0.249186	0.655823	0.553213
4	AMD	0.160818	0.435198	0.186155	0.589695	0.496193
5	APPS	0.291426	0.350243	0.072717	0.587222	0.439339
6	ASML	0.524659	1.000000	0.596358	0.000000	0.000000
7	AVGO	0.757775	0.721690	0.374722	0.924004	0.362481
8	AZPN	0.202995	0.515103	0.199674	0.597551	0.553060
9	BIDU	0.261581	0.448621	0.148525	0.596716	0.509801
10	BR	0.662223	0.155535	0.095317	0.645781	0.314629

Figure 8: Part of the Normalized Data.

0, the better the result. We will deduct all the points by 1, and since the closer the data is to 0, the better the stock, that's why the reference point later will be 0. The closer to the point, the better the stock.

Then we do the PCA process, according to our methodology part and we are able to draw the following graph:

A lot of information can be deduced from this graph. As we mentioned in the methodology part, the closer the stock is to the reference point, the better it will be. From the chart, the investor should buy stocks: 'BR', 'APPL', 'IDDX'. In fact, we can calculate the distance between the stock point and the reference point and set a standard to tell the 'close'. This can be done in the future modification of this project.

From the graph we can validate an assumption of a really basic but famous and useful model, the Capital Asset Pricing Model: investors have the same expectation to the market. In the graph we see that huge amount of points are clustered together. It means that their prices are influenced by these two main components similarly, which means that in most of the time investors will react to the same information in the somewhat same way.

5 Authentication of Methods

When we write the first version at April 26, we recommend three stocks to long. Apple, BR, and IDX. Today, at May 7th, our methods proves valid. All three stocks prices rise a lot. It is shown in the following figures.
Apple stock price rises from 163.76 dollar to 173.57 dollar, 6 percent rise in 2 weeks.

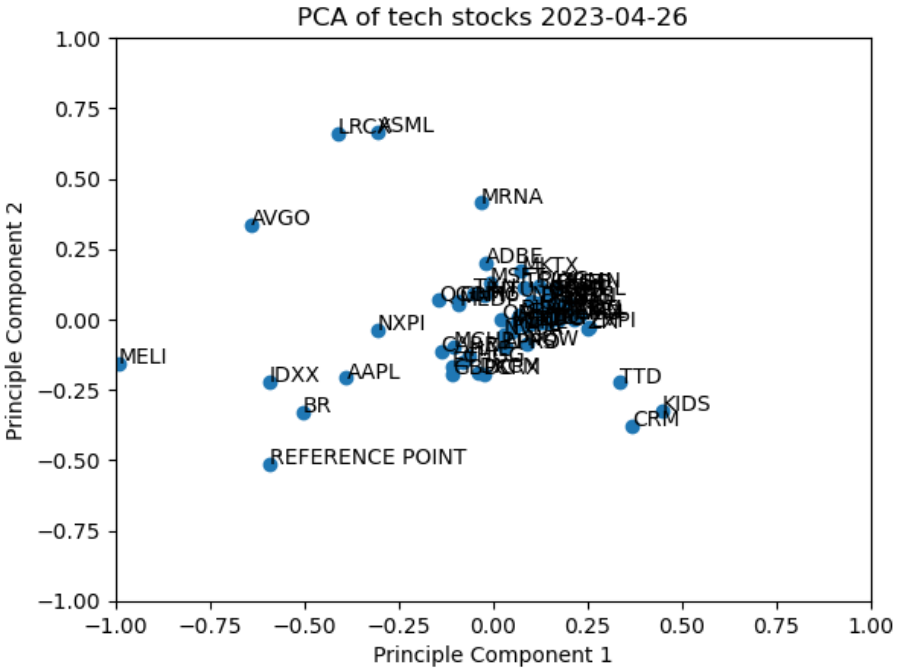

Figure 9: PCA result

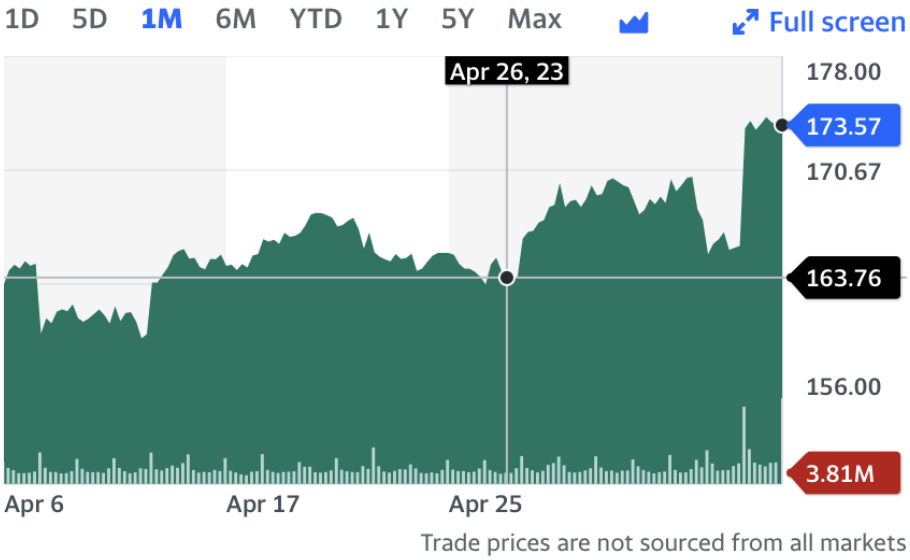

Figure 10: Apple stock price

BR stock price rises from 140.04 dollar to 152.87 dollar, 9.16 percent rise in 2 weeks.

Figure 11: BR stock price

IDX stock price rises from 18.13 dollar to 18.30 dollar, 1 percent rise in 2 weeks.

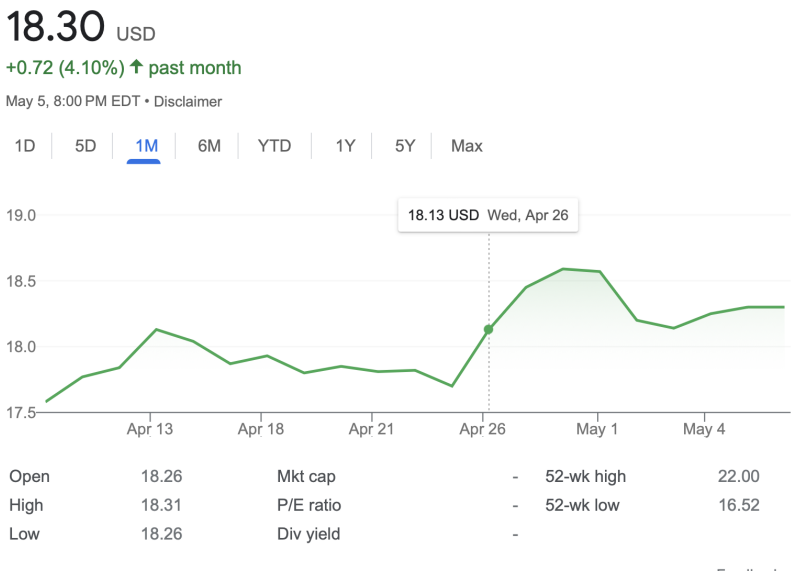

Figure 12: IDX stock price

6 Conclusion

In conclusion, PCA is a powerful tool for dimensional reduction and extraction of key factors from a large data set. However, for the datasets with a few features, it might not seem so useful, since analyzing all these features is not tough work for the computer. On the other hand, for datasets with thousands of features, it is very useful, since analyzing such large features costs lots of memory space and runtime. In this case, the PCA could play the best role on analyzing the large-size data.

After normalizing the data and establishing the reference points, we performed PCA to transform the data set into a principal component space. This allowed us to identify the most important factors contributing to the variance in the data.

Investors can find inexpensive stocks that are worthwhile buying by examining stocks in the major component space. By encouraging the return of stock prices to their intrinsic values, this helps both the rational investor and the market as a whole. It is crucial to remember that the criteria(factors we are using to do PCA in this case) chosen might have an impact on the outcomes, and investors may decide to take additional elements into account depending on their tastes and investing plans.

Overall, PCA is a valuable technique for stock market analysis, allowing investors to make more informed decisions and take advantage of potential mispricings in the market.

7 Appendix

Code for this project:

```
import numpy as np
from scipy.linalg import svd

import numpy as np
from numpy import mean, std
from scipy.linalg import svd

def pca(data):
    # Define a matrix
    A = np.array(data)

    # Calculate the mean of each column
    M = mean(A.T, axis=1)

    # Calculate the standard deviation of each column
    S = std(A.T, axis=1, ddof=1)

    # Standardize the data by subtracting column means and dividing by standard
    X = (A - M) / S

    # Calculate covariance matrix of standardized matrix
    C = X.T.dot(X) * (1 / (X.shape[0] - 1))

    # Perform Singular Value Decomposition (SVD) on covariance matrix
    V, s, V_T = svd(C, full_matrices=False)

    # Project data
    pca = X.dot(V)

    # Calculate eigenvalues from the singular values
    eigenvalues = s**2 / (X.shape[0] - 1)

    # Calculate total variance
    total_variance = np.sum(eigenvalues)

    # Calculate explained variance ratio
    explained_variance_ratio = eigenvalues / total_variance

    return pca, explained_variance_ratio

# Set the desired minimum and maximum values for the scaled data
min_val = 0
```

```
max_val = 1

# Normalize data between 0 and 1
def normalization(column):
    col_std = (column - column.min()) / (column.max() - column.min())
    col_scaled = col_std * (max_val - min_val) + min_val
    return col_scaled
# Extract today's data
# Define a list of stocks in technology sector
stocks = ['AAPL', 'ABCL', 'ABNB', 'ADBE','AMD',
          'APPS', 'ASML', 'AVGO','AZPN', 'BIDU',
          'BR', 'CARR', 'CDNS', 'CHGG', 'CRM', 'CSCO',
           'DLO', 'DOX', 'DXCM', 'ET', 'EXEL', 'EXPI', 'FLGT',
           'FUTU', 'GBDC', 'GGG', 'GLOB', 'GNRC', 'GOOGL', 'GRMN',
           'HAE', 'HLNE', 'IDXX', 'INTC', 'INTU', 'KIDS',
          'LOGI', 'LPRO', 'LRCX', 'MCHP','MDRX', 'MDT', 'MEDP', 'MELI',
            'MKTX', 'MRNA', 'MSFT', 'MU', 'NOW', 'NTES', 'NVDA',
           'NXPI', 'OLED', 'OLLI', 'ON', 'PAYC', 'PCRX',
           'PYPL', 'QCOM', 'SEDG', 'TSLA', 'TTD', 'TXN', 'ZM']


# Create an empty list to store ratios data for each stock
pe_ratios = []
pes_ratios = []
de_ratios = []
pb_ratios = []
pr_ratios = []

# Loop through each stock in the list
#and retrieve its ratios from Yahoo Finance
for stock in stocks:
    ticker = yf.Ticker(stock)
    pe_ratio = ticker.info['trailingPE']
    pes_ratio = ticker.info['trailingEps']
    de_ratio = ticker.info['debtToEquity']
    pb_ratio = ticker.info['priceToBook']
    pr_ratio = ticker.info['fiftyTwoWeekHigh'] -
               ticker.info['fiftyTwoWeekLow']

    pe_ratios.append(pe_ratio)
    pes_ratios.append(pes_ratio)
    de_ratios.append(de_ratio)
    pb_ratios.append(pb_ratio)
    pr_ratios.append(pr_ratio)
```

```
# Write ratios data to a CSV file
with open('technology_stock_5ratios.csv', mode='w') as file:
    writer = csv.writer(file)
    writer.writerow(['Stock',
                     'Price_to_Earnings',
                     "Earnings_Per_Share",
                     "Dept_to_Equity", 'Price_to_Book',
                     'Price_to_52W_Range'])
    for i in range(len(stocks)):
        writer.writerow([stocks[i], pe_ratios[i],
                         pes_ratios[i], de_ratios[i],
                         pb_ratios[i], pr_ratios[i]])

df = pd.read_csv('technology_stock_5ratios.csv')
df.set_index("Stock")

# Apply normalization to all columns except the first one
df.iloc[:, 1:] = df.iloc[:, 1:].apply(normalization)
normalized_data = df

# Add reference point
new_row = pd.DataFrame([['REFERENCE_POINT', 0, 1, 0, 0, 0]],
        columns=normalized_data.columns)

# Add the new row using pandas.concat
normalized_data = pd.concat([normalized_data, new_row],
                    ignore_index=True)


modified_data = normalized_data.copy()

# subtract normalize data from 1
modified_data.iloc[:, [1, 3, 4, 5]] =1 -
    normalized_data.iloc[:, [1, 3, 4, 5]]

modified_data

# stock name
stocks = modified_data.iloc[: , 0]
# stock data
data = modified_data.iloc[: , 1 :]
# pca
reducedData, explained_variance_ratio = pca(data.values)

Comp1 = reducedData[:, 0]
Comp2 = reducedData[:, 1]
```

```
df = pd.DataFrame(reducedData, columns=['Comp1', 'Comp2',
    'Comp3', 'Comp4', 'Comp5'])
df.insert(0, 'Stock', stocks)
frame = df
frame.set_index("Stock")

plt.scatter(Comp1, Comp2)
current_date = datetime.now().date()
plt.title('PCA of tech stocks ' + str(current_date))
plt.xlabel('Principle Component 1')
plt.ylabel('Principle Component 2')

for i in range(len(stocks)):
    plt.annotate(stocks[i], (Comp1[i], Comp2[i]))

plt.xlim(-1, 1)
plt.ylim(-1, 1)
plt.show()
```