

Final Project

Team Members: Kaile Huang; Jinhong Lei; Bingxu Chen

Abstract

The stock market is an extremely complicated system, including high noise, non-linear data and massive reasonless investors. These factors made the stock price and investing process are very hard to forecast. Thus, the problem of deciding which industry to invest is becoming a more and more difficult decision-making progress for investors. It's a very hard problem and the solution is always full of fuzzy and always could not be constantly precise. This paper mainly constructs the model based on PCA algorithm and K-Means algorithm as well as Decision Tree algorithm to decide which industry to invest in bull market (good market performance) or to avoid in bear market (bad market performance). Through the analysis comes from the real data and strict model, the evaluation and conclusion may still have the bias and timeliness since the economical tendency is changing very fast in recent years and thus the parameters in the model also need to modify according to the economical trend sometimes.

We decide to use PCA model to reduce the dimensions of data we found from two professional APIs and then use the processed weekly data each company to build the K-Means model. The meaning of doing this is to divide these companies into four clusters, which are high-yield and high-volume cluster, high-yield but low-cluster, low-yield but high-volume cluster and low-yield and low-volume cluster. That could help us to find which industries should we invest in bull market and which industries should we avoid in bear market. Finally, we picked up some company with good comprehensive performance in its industry to test the result and analysis we gained above by using the Decision Tree algorithm. We want to find out if our results could be utilized in the real world. If so, the paper is qualified the ability to provide investment strategy and opinions. In the specific research process, the data in this paper is mainly selected from the Nasdaq stock exchange. Therefore, the cluster model may only describe the characteristics of stocks with different good or bad performance and provide strategy in Nasdaq stock market.

Introduction and Related Work

Accompanied with the rapid development of data science, many aspects of all walks in life are gradually in a tangle with data, especially the finance industry. Furthermore, the stock market contains huge amount of data including the company data, trading data, investment data, economic data, etc. Thus, there is a more and more strong strength for investor to use these data to make decision in so many ways by using different of qualitative math and computer tools like model. Since the core law of evaluation is computing by different models to judge an investment, the models from data science take a more and more important part in the investing area.[Benjamin Graham, 1935] Compared with the traditional methods of evaluation based on mathematical analysis, machine learning model has the more advantages on implementing huge amount of data and dealing much more information and find more strong and precise factors that influence the asset prices.[Audretsch DB, 2006] For example, you may want to reduce many redundancy factors into several precise factors that influence deeply on stock trading, then the

PCA model could help you but traditional methods could do nothing. Furthermore, analysis of the listed company financial information requires analyst to divide them into several clusters so that they could deal with them better. However, traditional methods could only use property of these indicators to judge which indicators should be divide into one cluster. Actually, with the help of K-Means model or other clustering models we have learned from the machine learning classes, we can easily deal with such problems and the result could make more sense. Also, we can use decision tree to assist us formulating some investing strategies thus we can save a lot of energy and money on some pattern that repeated in massive times. Therefore, it is of great theoretical and practical significance to use these technologies to complete your investing strategy. [A. Raharto Condrobimo, 2016]

The development of stock investment theory in western world is much more mature than that in China. Many western scholars have used machine learning technology to conduct investment analysis and built qualitative models through various algorithms to make prediction or explain the impact on macroeconomic as well as the impact on specific sectors.[Fama EF, 1993] Beat Wuthrich used data mining technology to design the stock forecast and analysis system to predict the future trend of the stock market.[Wuthrich, 1998] G. Peter Zhang studied the application of ARIMA model and neural network model on stock analysis with time series data, and the results show that the application of neural network model on stock data prediction is more obvious.[G.Peter Zhang, 2003] David used the neural network model to predict the comprehensive return of stocks, and the results show that the classification model of neural network is better than other models to form investment decisions with higher returns under the same risk. To predict when the listed company's share price growth trend Refenes researchers carried on the comparative analysis through the traditional multiple linear regression analysis and neural networks.[Refenes A, 1994] The results showed that the neural network's characteristics have better performance on data fitting aspect and were able to show a much higher precision compared with the traditional stock prediction analysis methods.

The Origin of PCA, K-Means and Decision Tree Algorithm

PCA:

It is usually necessary to observe the data containing multiple variables and collect a large amount of data for analyzing in many fields. Multivariate large data sets will undoubtedly provide abundant information for research and application. However, they also increase the workload of data collection in some extent. What's more, in many cases, there may be correlations between many variables, which increases the complexity of the analysis. If each indicator is analyzed separately, the analysis is often isolated and the information in the data cannot be fully explained. Therefore, blindly reducing the indicators will lose a lot of useful information and lead to wrong conclusions.

Therefore, it is necessary to find a reasonable method to minimize the loss of information contained in the original index while reducing the indicators to be analyzed thus we could achieve the purpose of comprehensive analysis of the collected data. Therefore, we need to

transfer some closely related variables into some new variables so that these new variables are unrelated. Principal component analysis is one of this kind of dimension reduction algorithm.

PCA (Principal Component Analysis) is one of the most widely used data dimension reduction algorithms. The main idea of PCA is to map the n -dimensional features to the k -dimensional features, which is a new orthogonal feature also known as the principal component. It is the k -dimensional features reconstructed on the basis of the original n -dimensional features. The work of PCA is to find a set of orthogonal coordinate axes sequentially from the original space, and the selection of the new coordinate axes is closely related to the data itself. Among them, the selection of the first new coordinate axis is the direction with the largest difference in the original data, the selection of the second new coordinate axis is the one with the largest variance in the plane orthogonal to the first coordinate axis, and the selection of the third axis is the one with the largest difference in the plane orthogonal to the first and second axes. And by analogy, you get n axes like this. With the new axes obtained in this way, we find that most of the variance is contained in the first k axes, and the variance in the rear axes is almost zero. So, we can ignore the rest of the axes and just keep the k axes that have most of the variance. In fact, this is equivalent to only retaining the dimensionality features that contain most of the variance, while ignoring the dimensionality features that contain almost zero variance, so as to realize the dimensionality reduction processing of data features. [Jeng AM., 2016]

K-Means:

K-means algorithm is a basic classification algorithm for the number of known clustering categories. It is a typical distance-based clustering algorithm, which uses distance as the evaluation index of similarity. That means the closer the distance between two objects is, the greater the similarity will be. The algorithm considers that the cluster is composed of objects close to each other, so the final target is to get a compact and independent cluster. It is measured using Euclidean distance.

The k-means algorithm first randomly selects k points as the initial clustering center, then calculates the distance between each data object and the clustering center and classifies the data object into the class where the nearest clustering center is located. The adjusted new class calculates the new clustering center. If there is no change in the clustering center adjacent to each other, it means the data object adjustment is over and the clustering criterion function f has converged. In each iteration, the classification of each sample should be examined to see if it is correct. If it is not, it should be adjusted. After adjusting all the data, modify the clustering center and enter the next iteration. If all data objects are correctly classified in an iterative algorithm, there will be no adjustment and no change in the clustering center, which indicates that f has converged, and the algorithm is over.

Decision Tree:

Decision tree is a classification algorithm of supervised learning methods in machine learning. This prediction model represents a mapping between object properties and object values. Each node in the tree represents an object, each path represents a possible property value, and each leaf represents the value of the object represented by the path from the root node to the leaf node.

Decision tree model has only a single output. It actually provides a rule-like way of getting values under what conditions.

Intuitively, the decision tree classifier is like a flowchart consisting of a judgment module and a stop block, which represents the classification result (that is, the leaves of the tree). The judgment module represents the determination of the value of a feature (if the feature has several values, the judgment module has several branches).

If efficiency is not taken into account, all the features of the sample will eventually be cascaded together, and a sample will be divided into a class termination block. In fact, among all the characteristics of the sample, some of them play a decisive role in classification. The process of constructing a decision tree is to find these decisive features and then construct an inverted tree according to their determinateness. The most decisive characteristic acts as the root node, and then recursively finds the next most decisive characteristic in the subsets of each branch until all the data in the subsets belong to the same category. Therefore, the process of constructing a decision tree is essentially a recursive process of classifying data sets according to their characteristics. [Liu, G. 2011]

The Problem We Proposed

We want to design a PCA model to implement the multi-dimensional data into two-dimensional data, which is price factor and volume factor. Thus, we can avoid the effects from the similar factors so that makes the result more reasonable and precisely. The data structure before and after we processed are shown below:

Before:

Index	Type	Size	
0	str	1	{"CMCSA": {"2020-03-27": [33.165, 37.31, 31.705, 34.57, 209170342.0], .
1	str	1	{"INTC": {"2020-03-27": [46.02, 55.95, 45.9132, 52.37, 210933187.0], "
2	str	1	{"ALTR": {"2020-03-27": [26.22, 27.31, 24.26, 24.96, 2243246.0], "2020
3	str	1	{"ALXN": {"2020-03-27": [81.89, 87.0, 77.2275, 84.01, 14157639.0], "20
4	str	1	{"ADP": {"2020-03-27": [108.53, 138.14, 104.89, 131.38, 20659389.0], "
5	str	1	{"TILE": {"2020-03-27": [6.01, 7.48, 5.65, 7.11, 4533910.0], "2020-03-

After:

Key	Type	Size	
2019-11-01	list	2	[4.726071872264013, 81783135.0]
2019-11-08	list	2	[4.2993697391807375, 86219761.0]
2019-11-15	list	2	[4.547149366106987, 69370451.0]
2019-11-22	list	2	[3.722300821270638, 78335499.0]
2019-11-29	list	2	[3.3465919922879133, 93763688.0]

Scripts that we may want to explain

Then we need to process these two-dimensional data for latter use. For each company, we use every price in this week minus the price in the last week and divides the price of this week. Basically, as the formula below:

$$\text{Gaining Ratio} = (\text{Price}_t - \text{Price}_{t-1}) / \text{Price}_t$$

Then we use K-Means model to deal with our data matrix weekly since we want to make 4 different clusters every week according to the two dimensions. That means we have 4 clusters with different characteristics, which are high-yield and high-volume cluster, high-yield but low-volume cluster, low-yield but high-volume cluster and low-yield and low-volume cluster. We need to use the high-yield and high-volume cluster in the bull market and low-yield but high-volume cluster in the bear market. That's because we need to buy high-yield stock in bull market to gain more and sell the low-yield stock in bear market to avoid lose. The high-volume is convenient for trading since low-volume stock means less people are participating the trading. Finally, we could find the following investing strategies for example:

Industries are suitable for holding in the bull market:

Energy; Financials; Materials; Communication Services; Information Technology.

Industries are suitable for avoiding in the bear market:

Consumer Discretionary; Industrials; Real Estate; Utilities; Health Care; Consumer Staples.

After above analysis, we still need to test if our judging is correct. That's where we use the Decision Tree model. We could create a data structure as below by using the results we gained above. The Decision Tree model could assist us predicting the market performance when we know the performance of some sectors in certain periods. (Y means Yes, N means No, U means Unobvious)

Week	Energy	Financials	Materials	Commu Serv	Info Tech	Consum Discre	Industrials	Real Estate	Utilities	Health Care	Consum Stapl	Market Perform
1	Y	Y	Y	Y	Y	Y	U	N	N	U	Y	BULL
2	N	U	U	Y	Y	Y	U	Y	Y	Y	Y	BULL
3	U	Y	U	U	N	N	U	N	Y	Y	Y	BULL
4	N	Y	N	Y	Y	U	U	Y	N	Y	U	BULL
5	Y	U	U	Y	U	N	N	U	N	U	N	BULL
6	U	N	Y	N	Y	U	N	N	N	N	U	BULL
7	Y	Y	U	Y	Y	Y	Y	Y	Y	Y	Y	BULL
8	U	U	N	U	Y	U	N	U	N	U	U	BULL
9	U	Y	Y	Y	Y	U	Y	Y	Y	Y	Y	BEAR
10	U	U	U	Y	Y	Y	U	Y	U	U	N	BEAR
11	U	U	U	N	N	N	U	U	Y	U	U	BEAR
12	N	N	N	N	N	N	N	N	N	N	N	BEAR
13	N	N	Y	N	U	N	N	Y	Y	U	Y	BEAR
14	N	N	N	N	N	N	N	N	N	N	N	BEAR
15	N	N	N	N	N	U	N	N	N	U	Y	BEAR
16	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	BEAR

Experimental Results

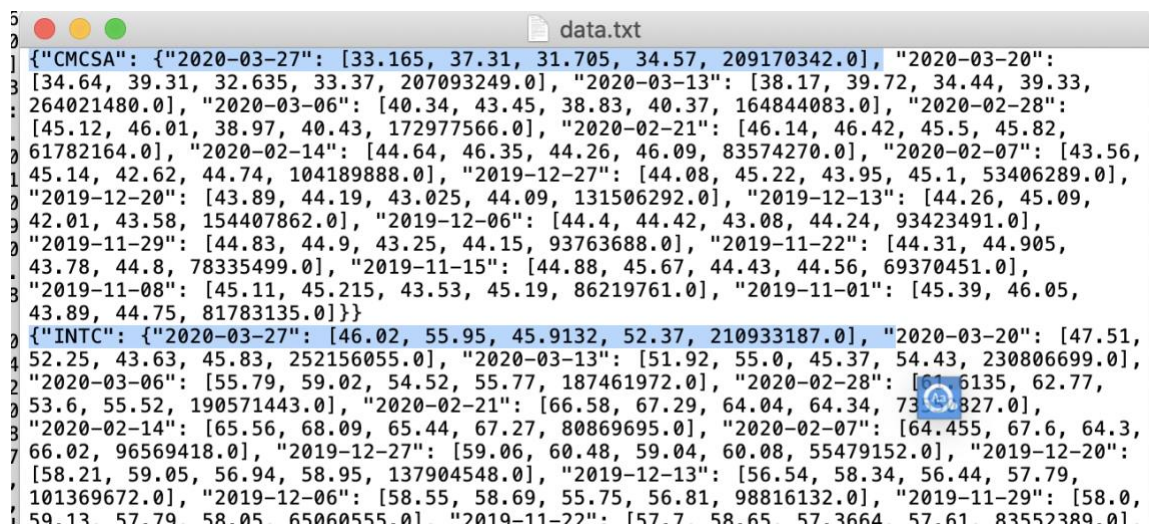
Scripts that we may want to explain

PCA

According to the data we scraped from the internet, we found that there are four different prices of each stock. Although all of these four prices are meaningful, applying all of them is very redundant and that will make problems more difficult, so we tried to use principle components analysis algorithm to turn four prices into one.

The picture below is part of the original data we have. It's a dictionary of dictionary, the outside key is the name of the company, and value is an also a dictionary. In the value dictionary, keys are the dates when the data is recorded, the frequency is about once a week,

The value is a list that contains 4 different prices and the last one is the trading volume; we want to transfer this list that only contains one price and one trading volume.

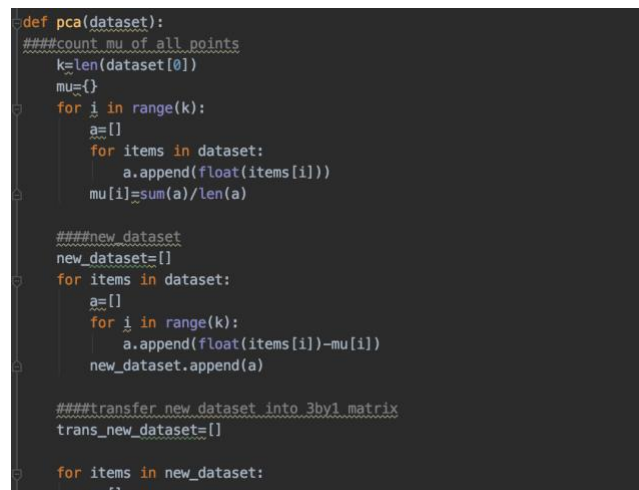


```

{"CMCSA": {"2020-03-27": [33.165, 37.31, 31.705, 34.57, 209170342.0], "2020-03-20": [34.64, 39.31, 32.635, 33.37, 207093249.0], "2020-03-13": [38.17, 39.72, 34.44, 39.33, 264021480.0], "2020-03-06": [40.34, 43.45, 38.83, 40.37, 164844083.0], "2020-02-28": [45.12, 46.01, 38.97, 40.43, 172977566.0], "2020-02-21": [46.14, 46.42, 45.5, 45.82, 61782164.0], "2020-02-14": [44.64, 46.35, 44.26, 46.09, 83574270.0], "2020-02-07": [43.56, 45.14, 42.62, 44.74, 104189888.0], "2019-12-27": [44.08, 45.22, 43.95, 45.1, 53406289.0], "2019-12-20": [43.89, 44.19, 43.025, 44.09, 131506292.0], "2019-12-13": [44.26, 45.09, 42.01, 43.58, 154407862.0], "2019-12-06": [44.4, 44.42, 43.08, 44.24, 93423491.0], "2019-11-29": [44.83, 44.9, 43.25, 44.15, 93763688.0], "2019-11-22": [44.31, 44.905, 43.78, 44.8, 78335499.0], "2019-11-15": [44.88, 45.67, 44.43, 44.56, 69370451.0], "2019-11-08": [45.11, 45.215, 43.53, 45.19, 86219761.0], "2019-11-01": [45.39, 46.05, 43.89, 44.75, 81783135.0]}}, {"INTC": {"2020-03-27": [46.02, 55.95, 45.9132, 52.37, 210933187.0], "2020-03-20": [47.51, 52.25, 43.63, 45.83, 252156055.0], "2020-03-13": [51.92, 55.0, 45.37, 54.43, 230806699.0], "2020-03-06": [55.79, 59.02, 54.52, 55.77, 187461972.0], "2020-02-28": [61.6135, 62.77, 53.6, 55.52, 190571443.0], "2020-02-21": [66.58, 67.29, 64.04, 64.34, 7311827.0], "2020-02-14": [65.56, 68.09, 65.44, 67.27, 80869695.0], "2020-02-07": [64.455, 67.6, 64.3, 66.02, 96569418.0], "2019-12-27": [59.06, 60.48, 59.04, 60.08, 55479152.0], "2019-12-20": [58.21, 59.05, 56.94, 58.95, 137904548.0], "2019-12-13": [56.54, 58.34, 56.44, 57.79, 101369672.0], "2019-12-06": [58.55, 58.69, 55.75, 56.81, 98816132.0], "2019-11-29": [58.0, 59.13, 57.79, 58.05, 65060555.0], "2019-11-22": [57.7, 58.65, 57.3664, 57.61, 83552389.0]}},

```

In order to achieve this target, we used the PCA function that wrote by ourselves.



```

def pca(dataset):
    """count mu of all points"""
    k=len(dataset[0])
    mu={}
    for i in range(k):
        a=[]
        for items in dataset:
            a.append(float(items[i]))
        mu[i]=sum(a)/len(a)

    """new dataset"""
    new_dataset=[]
    for items in dataset:
        a=[]
        for i in range(k):
            a.append(float(items[i])-mu[i])
        new_dataset.append(a)

    """transfer new dataset into 3by1 matrix"""
    trans_new_dataset=[]

    for items in new_dataset:
        a=[]

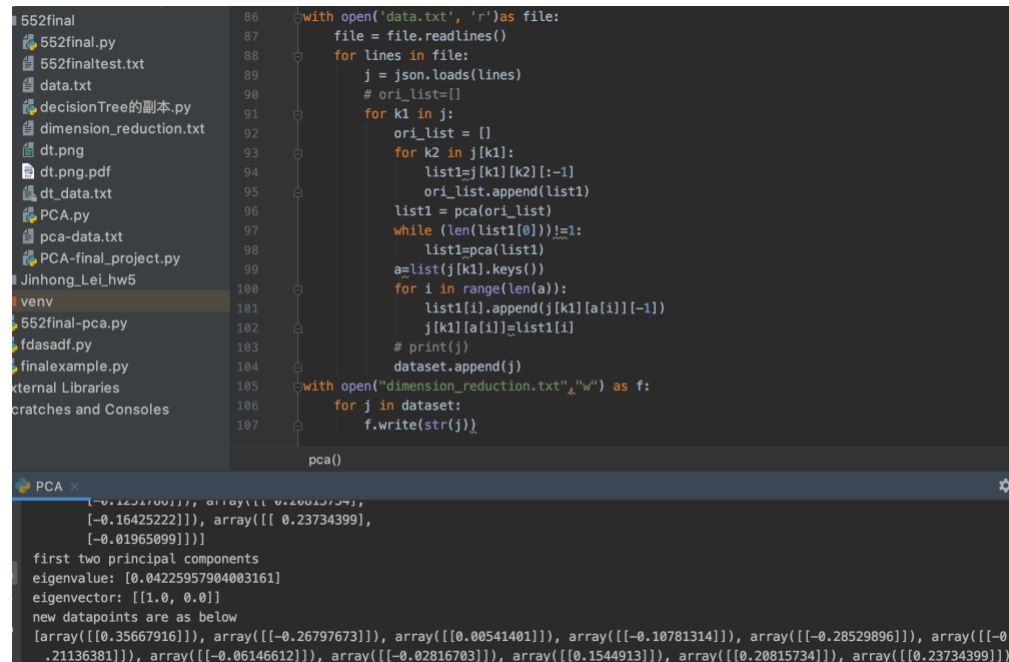
```

This PCA function takes a list of multidimensional coordinates which are different prices and make a dimensionality reduction, since each PCA can only reduce one dimension, so we might

Scripts that we may want to explain

552 Final Project

execute this function three times to get only one comprehensive price. In our function, we also print eigenvalue and eigenvector



```
552final
552final.py
552finaltest.txt
data.txt
decisionTree的副本.py
dimension_reduction.txt
dt.png
dt.png.pdf
dt_data.txt
PCA.py
pca-data.txt
PCA-final_project.py
Jinhong_Lei_hw5
venv
552final-pca.py
fdasadf.py
finalexample.py
External Libraries
Scratches and Consoles

86 with open("data.txt", 'r') as file:
87     file = file.readlines()
88     for lines in file:
89         j = json.loads(lines)
90         # ori_list=[]
91         for k1 in j:
92             ori_list = []
93             for k2 in j[k1]:
94                 list1=j[k1][k2][:-1]
95                 ori_list.append(list1)
96             list1 = pca(ori_list)
97             while (len(list1[0]))!=1:
98                 list1=pca(list1)
99                 a=list(j[k1].keys())
100                 for i in range(len(a)):
101                     list1[i].append(j[k1][a[i]][-1])
102                     j[k1][a[i]]=list1[i]
103                 # print(j)
104                 dataset.append(j)
105 with open("dimension_reduction.txt","w") as f:
106     for j in dataset:
107         f.write(str(j))

pca()

PCA
[[-0.12317001]], array([[ 0.26013734],
[-0.16425222]]), array([[ 0.23734399],
[-0.01965099]])]
first two principal components
eigenvalue: [0.04225957904003161]
eigenvector: [[1.0, 0.0]]
new datapoints are as below
[array([[0.35667916]]), array([[[-0.26797673]]), array([[0.00541401]]), array([[[-0.10781314]]), array([[[-0.28529896]]), array([[[-0.21136381]]), array([[[-0.06146612]]), array([[[-0.02816703]]), array([[0.1544913]]), array([[0.20815734]]), array([[0.23734399]])]
```

K-Means

Given the data after dimensionality reduction from PCA, we can do clustering and find interesting patterns from various clusters.

As the data is relatively large, we used pyspark to do the data processing and find clusters. And the clustering algorithm we chose is K-Means. To achieve better performance, we used K-Means++ to initialize centroids. This algorithm can significantly reduce the number of iterations and speed up the clustering process.

```
def initialize():
    # kmeans ++
    centroids = {sample(coords, 1)[0]}
    while len(centroids) < K:
        min_dis = points.map(lambda p: (p, min([euclid(c, p) for c in centroids]))) \
            .sortBy(lambda x: x[1], ascending=False)
        centroids.add(min_dis.take(1)[0][0])

    return centroids
```

Scripts that we may want to explain

```

sc = SparkContext()
data = sc.textFile('dimension_reduction.txt').map(lambda x: dict(json.loads(x))) \
    .flatMap(calculate_diff_ratio).groupByKey().mapValues(lambda x: list(x)) \
    .filter(lambda x: len(x[1]) > 1).collectAsMap()

d = 2
K = 4

# print(len(data))
for date, value in data.items():
    symbols = dict()
    coords = list()
    print(date)
    for c in value:
        coords.append((c[0], c[1]))
        symbols[(c[0], c[1])] = c[2]

    membership = kmeans_by_spark(coords) # {point_coords : centroid_coords}
    print(membership)
    print('*****')
    draw(date, membership, symbols)

```

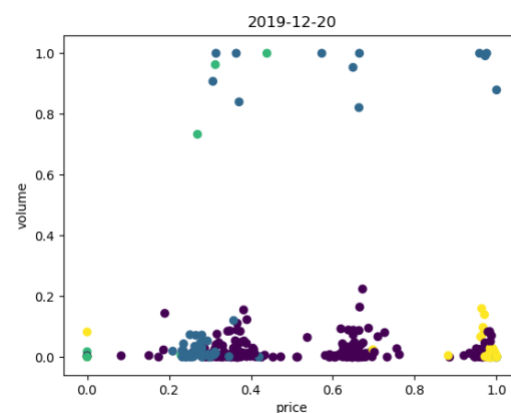
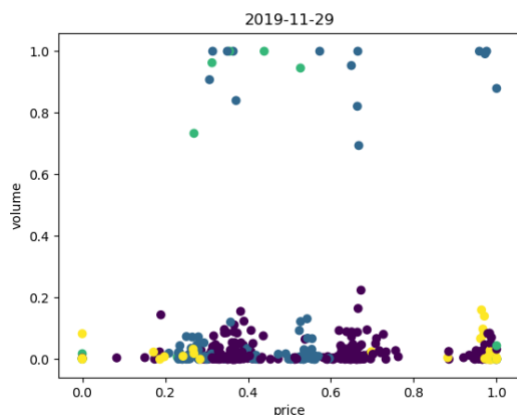
We also used the sklearn kmeans library to make sure our result is correct. And the clustering results are relatively same.

As we can see from the graphs below, we do cluster for the weekly stock data from November 2019 to December 2019 and from February 2020 to March 2020. The X-axis is price data, coming from PCA; and the Y-axis is volume data. As the two variables have different magnitudes, we did normalization on both variables using the formula $(X - X_{min}) / (X_{max} - X_{min})$. Thus, the measurement is more accurate and easier to drive conclusions.

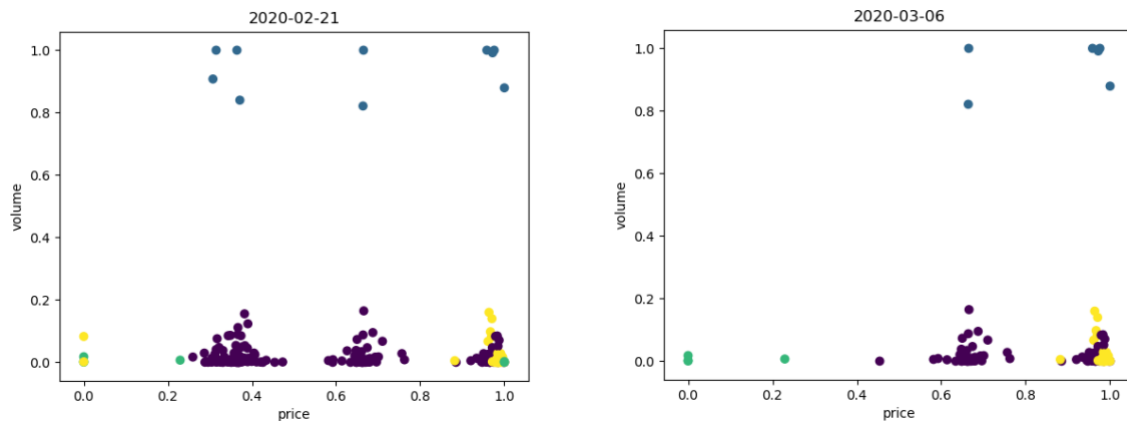
```

max_price = max([d[0][0] for d in pre_data])
min_price = min([d[0][0] for d in pre_data])
diff_price = max_price - min_price
max_volume = max([d[0][1] for d in pre_data])
min_volume = min([d[0][1] for d in pre_data])
diff_volume = max_volume - min_volume
training_data = [(d[0] - min_price) / diff_price, (d[1] - min_volume) / diff_volume for d in pre_data]

```



Scripts that we may want to explain



Decision Tree

According to the results from k-means, we can find the groups of companies in different clusters, and distinguish the companies based on their company categories, so that we can conclude how different kinds of companies perform in both bull market and bear market. The data we have is shown below.

```

Decision_tree_data.txt
(Energy, Financials, Materials, Communication Services, Information Technology, Consumer Discretionary, Industrials, Real Estate, Utilities, Health Care, Consumer Staples, Market Performance)

01:Y,Y,Y,Y,Y,U,N,N,U,Y,bull market
02:N,U,U,Y,Y,Y,U,Y,Y,Y,bull market
03:U,Y,U,U,N,N,U,N,Y,Y,Y,bull market
04:N,Y,N,Y,Y,U,U,Y,N,Y,U,bull market
05:Y,U,U,Y,U,N,N,U,N,U,N,bull market
06:U,N,Y,N,Y,U,N,N,N,N,U,bull market
07:Y,Y,U,Y,Y,Y,Y,Y,Y,Y,bull market
08:U,U,N,U,Y,U,N,U,N,U,U,bull market

09:U,Y,Y,Y,Y,U,Y,Y,Y,Y,Y,bear market
10:U,U,U,Y,Y,Y,U,Y,U,U,N,bear market
11:U,U,U,N,N,N,U,U,Y,U,U,bear market
12:N,N,N,N,N,N,N,N,N,N,N,bear market
13:N,N,Y,N,U,N,N,Y,Y,U,Y,bear market
14:N,N,N,N,N,N,N,N,N,N,N,bear market
15:N,N,N,N,N,U,N,N,N,U,Y,bear market
16:Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,Y,bear market

```

Since we have derived some critical information from the clustering algorithm and the dimensionality reduction algorithm, which are k-means and PCA, we want to use a decision tree to predict the stock market situation.

In the data that we get from formal steps, we can set the stock market performance as the label, including bull market and bear market, company categories as attributes including energy, financials, materials and so on.

We read this text file using the decision tree code wrote by ourselves, put the data into two parts, one is train_data, this is a nested dictionary, each key-value pair represents one attribute. The

Scripts that we may want to explain

value is also a dictionary containing each possible value of the attribute and its corresponding row numbers. The other one is a list of labels.

```
training data is {'Energy': defaultdict(<class 'list'>, {'Y': [0, 4, 6, 15], 'N': [1, 3, 11, 12, 13, 14], 'U': [2, 5, 7, 8, 9, 10]}), 'Financials': defaultdict(<class 'list'>, {'Y': [0, 2, 3, 6, 8, 15], 'U': [1, 4, 7, 9, 10], 'N': [5, 11, 12, 13, 14]}), 'Materials': defaultdict(<class 'list'>, {'Y': [0, 5, 8, 12, 15], 'U': [1, 2, 4, 6, 9, 10], 'N': [3, 7, 11, 13, 14]}), 'Communication Services': defaultdict(<class 'list'>, {'Y': [0, 1, 3, 4, 6, 8, 9, 15], 'U': [2, 7], 'N': [5, 10, 11, 12, 13, 14]}), 'Information Technology': defaultdict(<class 'list'>, {'Y': [0, 1, 3, 5, 6, 7, 8, 9, 15], 'N': [2, 10, 11, 13, 14], 'U': [4, 12]}), 'Consumer Discretionary': defaultdict(<class 'list'>, {'Y': [0, 1, 6, 9, 15], 'N': [2, 4, 10, 11, 12, 13], 'U': [3, 5, 7, 8, 14]}), 'Industrials': defaultdict(<class 'list'>, {'U': [0, 1, 2, 3, 9, 10], 'N': [4, 5, 7, 11, 12, 13, 14], 'Y': [6, 8, 15]}), 'Real Estate': defaultdict(<class 'list'>, {'N': [0, 2, 5, 11, 13, 14], 'Y': [1, 3, 6, 8, 9, 12, 15], 'U': [4, 7, 10]})
```

```
decision_tree
/Users/leijinhong/Desktop/pycharm/venv/bin/python /Users/leijinhong/Desktop/pycharm/552final/decision_tree.py
labels are ['bull market', 'bull market', 'bull market', 'bull market', 'bull market', 'bull market', 'bull market', 'bull market', 'bear market', 'bear market', 'bear market', 'bear market', 'bear market', 'bear market', 'bear market', 'bear market']
```

What's more, we also defined a class for tree nodes which have 4 attributes, we can use them to build a tree by recursively calling the generateDT function.

```
from collections import defaultdict, Counter
from math import log
from queue import Queue
from graphviz import Graph

class TreeNode(object):
    def __init__(self, name=None, rows=None, attributes=None, branches=None):
        self.name = name
        self.rows = rows
        self.attributes = attributes
        self.branches = branches
```

Conclusions

PCA

The output maintains the original format.

```
dimension_reduction.txt
Q> into
{'CMCSA': {'2020-03-27': [-17.061859553872875, 209170342.0], '2020-03-20': [-15.657733685158028, 207093249.0], '2020-03-13': [-9.574886072112655, 264021480.0], '2020-03-06': [-4.022004482405514, 164844083.0], '2020-02-28': [-0.5256973094552766, 172977566.0], '2020-02-21': [6.7381282978327155, 61782164.0], '2020-02-14': [5.345755217394466, 83574270.0], '2020-02-07': [2.704826800197937, 104189888.0], '2019-12-27': [3.9676312943855647, 53406289.0], '2019-12-20': [2.4454279928298464, 131506292.0], '2019-12-13': [2.0987571670487064, 154407862.0], '2019-12-06': [2.900170542204821, 93423491.0], '2019-11-29': [3.3465919922879133, 93763688.0], '2019-11-22': [3.722300821270638, 78335499.0], '2019-11-15': [4.547149366106987, 69370451.0], '2019-11-08': [4.2993697391807375, 86219761.0], '2019-11-01': [4.726071872264013, 81783135.0]}}{'INTC': {'2020-03-27': [-16.00889114827993, 210933187.0], '2020-03-20': [-21.15145903434, 252156055.0], '2020-03-13': [-12.681149703520955, 230806699.0], '2020-03-06': [-3.032122081350133, 187461972.0], '2020-02-28': [0.8300211591508532, 190571443.0], '2020-02-21': [15.535713504345283, 73550827.0], '2020-02-14': [17.5353940940592, 80869695.0], '2020-02-07': [15.504115392570903, 96569418.0], '2019-12-27': [3.947424202070526, 55479152.0], '2019-12-20': [1.1613825170274201, 137904548.0], '2019-12-13': [-0.830962707987919, 101369672.0], '2019-12-06': [-0.5178793875258731, 98816132.0], '2019-11-29': [1.1618166501691913, 65060555.0], '2019-11-22': [0.3563809496537467, 83552389.0], '2019-11-15': [0.7118423887469422, 66854444.0], '2019-11-08': [-0.07620217567539744, 86507450.0], '2019-11-01': [-2.4454254759952554, 102680287.0]}}{'ALTR': {'2020-03-27': [-14.779494000145327, 2243246.0], '2020-03-20': [-11.637644203188934, 2785391.0], '2020-03-13': [-9.439684989307821, 2197934.0], '2020-03-06': [-0.41540117765074663, 2391693.0], '2020-02-28': [3.312030371704779, 3280760.0], '2020-02-21': [8.395994017878307, 1554728.0], '2020-02-14': [10.758636516390688, 1051304.0], '2020-02-07': [10.253443780584943, 1199402.0], '2019-12-27':
```

Scripts that we may want to explain

As you can see, there are only one price and one trading volume in the value of data. We can use coordinates in this output text file to make clusters by k-means algorithm and do some further study.

In a word, PCA algorithm make our data more concise and simplify the follow-up research and reduce the running time of our codes, PCA can convert the original high-dimensional data vector into a lower-dimensional principal component, by doing this, we can get a better reconstructed data. In addition, the reconstructed data can still maintain the important information of the original version and even prettier than the original version to some extent.

K-Means

According to the process that we announced in the above, we calculated the statistic results of separate graphs from bull markets and bear markets. We only take companies which satisfy the price spread above 0.5 in the bull markets and companies which satisfy the price spread below 0.5 in the bear markets. Here is the graph in bull market and bear market:

Key	Type	Size	
AMCI Acquisition Corp.	list	2	['Finance', 'Business Services']
Acasti Pharma, Inc.	list	2	['Health Care', 'Major Pharmaceuticals']
Adial Pharmaceuticals, Inc	list	2	['Health Care', 'Major Pharmaceuticals']
Alexion Pharmaceuticals, Inc.	list	2	['Health Care', 'Major Pharmaceuticals']
Allogene Therapeutics, Inc.	list	2	['Health Care', 'Biotechnology: Biological Products (No Di']
Altair Engineering Inc.	list	2	['Technology', 'Computer Software: Prepackaged Software']
Alteryx Therapeutics Limited	list	2	['Health Care', 'Major Pharmaceuticals']
Automatic Data Processing, Inc.	list	2	['Technology', 'EDP Services']
Avis Budget Group, Inc.	list	2	['Consumer Services', 'Rental/Leasing Companies']
CYREN Ltd.	list	2	['Technology', 'Computer Software: Prepackaged Software']

Key	Type	Size	
icad inc.	list	2	['Health Care', 'Medical/Dental Instruments']
Veru Inc.	list	2	['Health Care', 'Major Pharmaceuticals']
USA Truck, Inc.	list	2	['Transportation', 'Trucking Freight/Courier Services']
TransGlobe Energy Corporation	list	2	['Energy', 'Oil & Gas Production']
The Cheesecake Factory Incorporated	list	2	['Consumer Services', 'Restaurants']
Sterling Bancorp, Inc.	list	2	['Finance', 'Savings Institutions']
Simmons First National Corporation	list	2	['Finance', 'Major Banks']
ShotSpotter, Inc.	list	2	['Technology', 'Computer Software: Prepackaged Software']
ShockWave Medical, Inc.	list	2	['Health Care', 'Medical/Dental Instruments']
Sesen Bio, Inc.	list	2	['Health Care', 'Major Pharmaceuticals']

Scripts that we may want to explain

Thus, our investment strategies are as below:

Industries are suitable for holding in the bull market:

Consumer Services; Finance; Health Care; Technology.

Industries are suitable for avoiding in the bear market:

Consumer Services; Finance; Health Care; Technology.

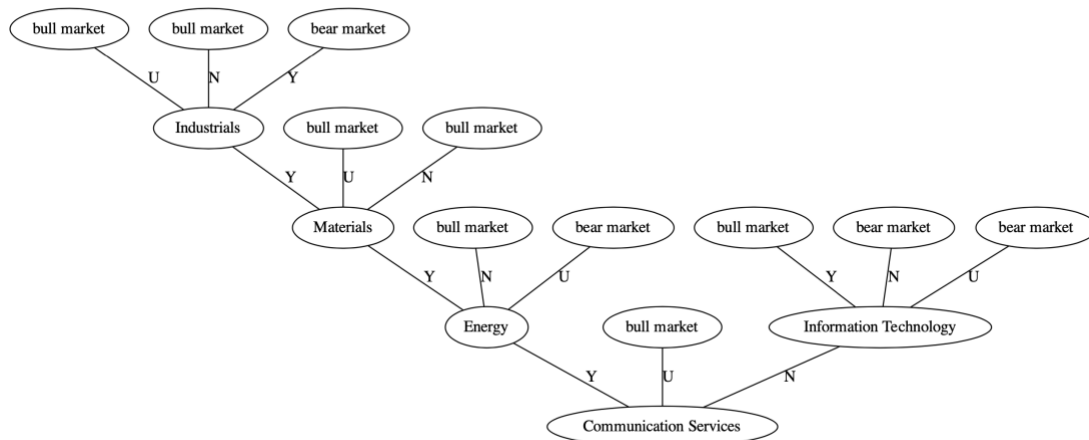
The two results are the same because these sectors are all belonging to the periodic industries thus very easy to fluctuate along with the market situation. Therefore, our research is right in the fundamental sense.

Then we use the specific sectors to calculate the return of investing and the results prove our conclusion in some extent. After we delete some extremely data points and make a weighted average value according to the volume of stocks weekly, the income of bull market is 56.20% and the income of bear market is – 63.53%.

total_gain	float 1	1.5620167102609772
total_loss	float 1	0.36469102170351736

Decision Tree

We also have a printDT function using graphviz library to draw decision tree and save it as a png file. The tree that our codes generated is shown below.



As far as we are concerned, this picture is a really meaningful decision tree, there is also a projection function in our codes for predicting, so that we can query the tree by giving it values of all different kinds of company and this tree will tell you the situation of the stock market, whether it is bull market or bear market.

Scripts that we may want to explain

```

170 test_data={"Energy": "Y", "Financials": "Y", "Materials": "Y", "Communication Services": "N", "Indu
171 "Consumer Discretionary": "Y", "Industrials": "N", "Real Estate": "U", "Utilities": "Y",
172 "Health Care": "Y", "Consumer Staples": "N"}
173 print('Prediction on test data: ' + projection(root, test_data))
174

```

decision_tree x

```

/Users/leijinhong/Desktop/pycharm/venv/bin/python /Users/leijinhong/Desktop/pycharm/552final/decision_tree.py
Prediction on test data: bear market

```

Contributions

Bingxu Chen grabs the raw data, which is the weekly stock price and volume from APIs and websites by using python and collect reports of three members then arrange them.

Jinhong Lei deals the raw data by using the PCA model to reduce the multi-dimensional data and uses the Decision Tree model to figure out the statement of stock market according to the performances of different sectors.

Kaile Huang builds the K-Means model by using the data that Jinhong Lei processed with PCA model and divides them into 4 clusters every week as well as analyzing the meaning behind the data clusters.

References

1. Graham B, Dodd DL. Security Analysis . New York: McGraw-Hill; 1935.
2. Audretsch DB, Keilbach MC, Lehmann E. Entrepreneurship and Economic Growth. Oxford: Oxford University Press; 2006.
3. Fama EF, French KR. Common risk factors in the returns on stocks and bonds. Journal of Financial Economics. 1993;33(1):3-56. doi:10.1016/0304-405X(93)90023-5
4. Wuthrich B, Cho V, Leung S, et al. Daily stock market forecast from textual Web data. In: PROC IEEE INT CONF SYST MAN CYBERN. Vol 3. 1998:2720-2725.
5. G. Peter Zhang. Time series forecasting using a hybrid ARIMA and neural network mode[J].Neuro computing.2003.(50):159-175.
6. Refenes A, Azema-Barac M. Neural network applications in financial asset management. Neural Computing & Applications. 1994;2(1):13-39. doi:10.1007/BF01423096
7. A. Raharto Condrobimo, Albert V. Dian Sano, Hendro Nindito. The Application Of K-Means Algorithm For LQ45 Index on Indonesia Stock Exchange. ComTech. 2016;7(2):151-159. doi:10.21512/comtech.v7i2.2256
8. Jeng AM. Using k-means and PCA in construction of a stock portfolio. 2016.
9. Li, T., & Liu, G. (2011). Stock Price's Prediction with Decision Tree. Applied Mechanics and Materials, 48-49, 1116–1121. <https://doi.org/10.4028/www.scientific.net/AMM.48-49.1116>