Equity Portfolio Optimization using Machine Learning Techniques
ECE/CS 498 Data Science Final Project
Spring 2020

Akhilesh Somani
somani4@illinois.edu
Mechanical Science and Engg, UIUC

Gowtham Kuntumalla
gowtham4@illinois.edu
Mechanical Science and Engg, UIUC

Manan Mehta
mananm2@illinois.edu
Mechanical Science and Engg, UIUC

***Abstract*—Personal Finance is an important aspect of any person in a professional career. Maintaining or improving the value of personal wealth can be done in multiple ways. To a common person, savings and investing in stock markets is the most approachable way. In this project, we discuss the concept of long term investing in stock market also commonly known as equity portfolio in a statistically optimized manner. We perform data exploration and custom feature generation for enhancing valuable domain knowledge. Then we display our experiments with clustering and predicting a particular stock price using recurrent neural network (with long short term memory blocks). We then build portfolios using Monte Carlo simulations and perform weights optimization to obtain return vs. return curves. Of note is the minimally correlated stocks selection obtained using orthogonal principal components analysis. Finally, we conclude that statistical analysis using historical price volume trend gives a valuable insight into future performance with associated risk. This acts a valuable tool in a savvy investor's due diligence repertoire.**

I. Introduction

Personal finance is an important aspect in every individual's career. Careful wealth management is crucial for long term financial health. Many individuals prefer to stay away from the stock market sometimes called as the equity market in fear of heavy losses. Instead, they prefer to place majority of their earnings in savings. Fig: 1 shows typical returns of $10,000 in a 10 year time frame. It assumes an average yearly return of $\approx$ 1.6% for savings account in US banks. Investing returns are shown for a typical stock portfolio mimicking S&P 500 index. Power of compounding is obviously a strong factor in favour of Investing.

From the same figure we observe that an investing account performs way better than a standard savings account in the US market. In the 10 years shown in figure, the initial capital invested in equity portfolio appreciates by $\approx$ 130% against $\approx$ 12% of savings account. If we take inflation into account then the returns for savings account will be nearly zero or even single digit negative percentage points.

Also, notice the slump in the value of the account around 2008. During this time, the US housing bubble burst due to banking sector taking excessive risk in the form of investing

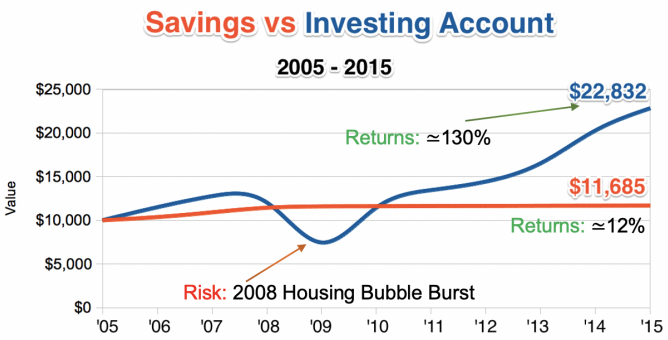

Fig. 1. Why Investing is better than Savings

in subprime collateralized debt obligation instruments (CDOs) in the mortgage market [1]. At times, the intricate links in the financial markets have severe negative effects on equity markets. Thus extraordinary returns are coupled with financial risks. This can be managed by cleverly balancing and creating a mixture of stocks. Often times, this volatility is used as an indicator of risk.

In this project, we primarily deal with equity market focusing on the stocks listed in the S&P500 index. It is a list of top 500 performing companies in a given time period as per market capitalization. It is compiled and updated by *Standard & Poors*, a popular credit rating agency. This introductory discussion boils down to solving a mathematical problem involving picking stocks and optimizing weights. It can be stated as following.

How to choose stocks in a portfolio to maximize returns while minimizing risk (commonly measured by volatility) over a fixed time frame?

A lot of work has gone into predicting future stock performance based on its past indicators. The goal of this study is to take the basic OHLCV (Open, High, Low, Close, Volume) data for S&P 500 stocks and perform statistical analysis using currently available data science techniques. We collected the

data used in this study from Kaggle [2].

Section II introduces various methods to explore the S&P 500 data-set. We employ methods such as Principal Component Analysis (PCA), and machine learning tools like Linear Discriminant Analysis (LDA), Clustering, Recurrent Neural Networks (RNNs), etc. We show our results in section 3. In section 4, we summarize our findings from data exploration and compare 3 different portfolios.

II. Methods

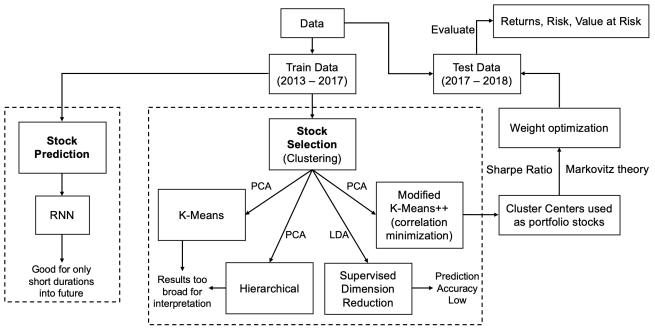

Fig. 2. Block diagram of solution methodology

In Fig: 2, we display all the different tactics we employed in reaching our final result.

The S&P 500 stock market index is maintained by S&P Dow Jones Indices and comprises of 500 common stocks issued by 500 large-cap companies and traded on American stock exchanges. The data covers the period from 2013 to 2018. OHLCV data for each stock is available on a daily basis.

A closer look at the data-set reveals that not all companies have data for all 5 years. For example, Alphabet was created in 2015, so its data is available only after its creation. Such stocks are removed from the analysis. 475 companies have data-sets for the entire 5-year period and only those are used for all analysis performed.

The initial task is to cluster the companies (in their respective sectors) based on key features (Feature Engineering). Identifying these features to efficiently perform clustering is a major challenge. Evaluation criterion for the efficacy of the generated features is comparison of predicted sectors with Global Industry Classification Standard (GISC).

Out of 111 features (technical indicators), 22 indicators are chosen after a careful study and understanding of what information these features provided. These technical features provide exhaustive information like momentum, volatility, trend, etc. which OHLCV values do not. The feature generation involves using these OHLCV values to generate these technical features. An entire list of the generated features is coded in the IPython files. As an example, the linear regression fit for Apple's stock (AAPL) is shown in Fig: 4

To perform clustering, the data needs to be transformed into a lower dimensional space. Principal Component Analysis (PCA) is performed on these newly generated features to reduce the number of features. The biggest challenge is trying

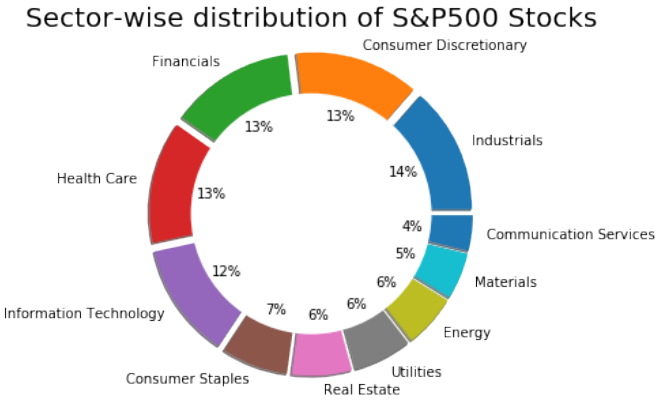

Fig. 3. GICS Sector Wise Distribution of S&P Stocks

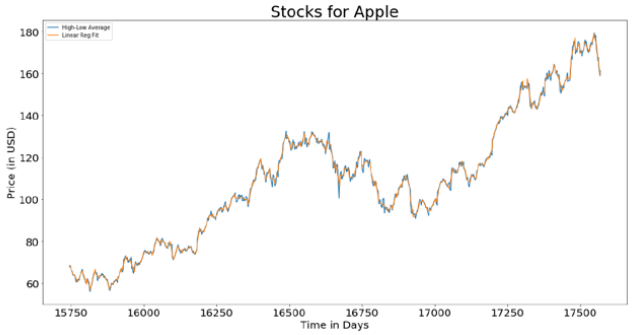

Fig. 4. Linear Regression Fit

to perform PCA on a 3D data-set, where one axis has features, another axis has company stocks, and the third axis has time-series data [3]. PCA expects a 2D data-set so it is necessary to reduce the space from 3D to 2D. Condensation of time-series data is performed by aggregating the time-series data into one single value representative of the entire series. Two parameters: mean and median are used to perform this aggregation. The time-granularity (i.e. the time-period chosen to condense the data) is varied to study the clusters observed. Only a couple of principal components are needed to explain over 85 variance in the data for all time-granularity cases.

PCA is performed on both scaled data (where the mean is zero and variance is one for all features) and un-scaled data (in the hopes of capturing the correlations between similar stocks). There are 11 sectors (as defined by GICS), and hence 11 clusters are expected to form. A KMeans clustering (using K = 11) yields clusters similar to the Fig: 5 in all the different cases (nature of data - unscaled v/s scaled, time granularity - yearly v/s 100 points v/s 50 points, measures to condense data - mean v/s median).

A closer analysis of the formed clusters by comparing with the actual sectors informs that companies within each KMeans cluster belong to varied sectors. The clusters formed by the KMeans clustering do not represent the actual sector wise distribution of the stocks. Kernel PCA do not yield the expected results too. (actual industrial sectors are obtained

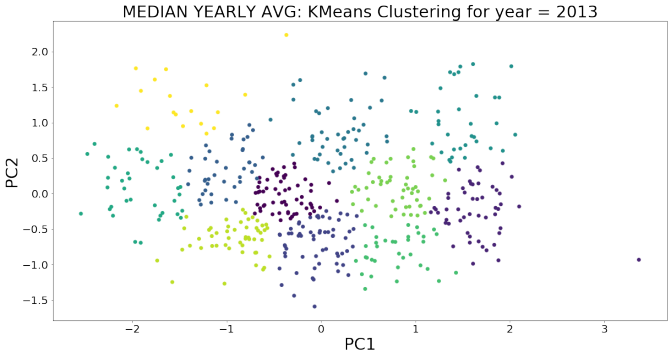

Fig. 5. KMeans Clustering for the year 2013 (Median as the measure for the condensed data)

from Wikipedia [4]).

As an alternative to PCA approach for dimensionality reduction, Linear Discriminant Analysis (LDA) is employed. LDA is a supervised way of reducing dimensions and achieving class separability. Data is divided into training and testing data-sets. Actual sectors for the training data are fed to LDA to achieve clustering. Testing data is used to predict the sectors and compare with the actual sectors. The average prediction accuracy of the sectors is just under 25% whereas the maximum prediction accuracy is around 32.5%. Fig: 6 shows the sector prediction accuracy of LDA over different samples, where each sample has 100 time-series data condensed to a single value - mean.

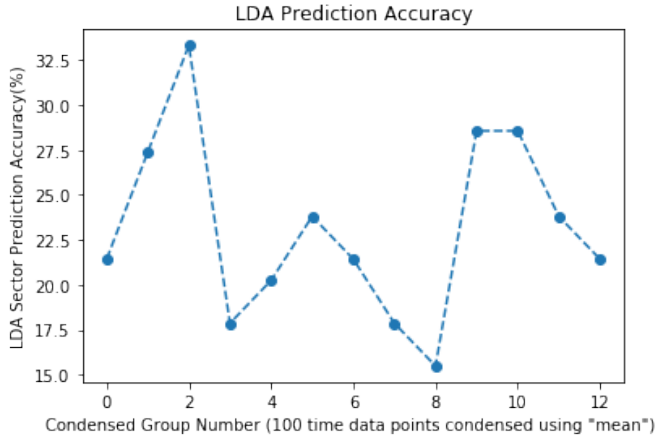

Fig. 6. LDA: Sector Prediction

One of the ways in which we can try investing is predicting how a stock will perform in the future and betting our money in that stock. We create a recurrent neural network using LSTM layers (Long Short Term Memory) popularised by their efficacy in the field of Natural Language Processing. It performs especially well for time series data [5], [6].

This model is trained using a grid search cross validation technique for multiple different hyper parameters. This 3 layer LSTM neural network performs well on the training data but falters in predicting long term future data. In the short term

Fig. 7. RNN-LSTM: Training AAPL data using historical OHLCV data 2013 - 2017

Fig. 8. RNN-LSTM: Predicting close price of AAPL from 2017 to 2018 at the beginning of 2017

this algorithm seems to approximately predict the nature of stock movement. Many blog articles available online falsely peek into the future data and get a wonderful looking test result which almost matches the actual test data. Abundant caution must be taken before trusting such articles. Possible reasons for poor performance of LSTMs are:

- Past prices of a company alone cannot predict long term future prices.
- Modeling complex inter company interactions by adding stock movements from different allied sectors can bring in additional information to improve prediction (computationally expensive). New information continually affects price significantly.
  - company news (in the figure above, Apple posted their quarterly earnings statement around trading day 50. This caused a massive spike in the share price)
  - Policy changes by the Governments.
  - Natural disasters, Pandemics etc.

Unless more information is given into the algorithm and appropriately complex model is used, there is no hope for

accurate long term predictions. This is going to be very costly and complicated due to the requirement of heavy computational power and myriad varieties of data.

In the light of these revelations, investing in a single stock is a risky business. A properly created portfolio typically has a lower risk associated with certain returns. For this purpose, we perform Monte Carlo simulations using non parametrical distribution of historical data.

The primary purpose of constructing a portfolio is to have a spread of stocks whose returns are not affected by each other, thereby reducing the risk from value loss in one or a few stocks. In an oversimplified manner, we say we want 'uncorrelated' stocks. However, different definitions of correlations between stocks have been deployed by financial experts, particularly for the equity market, the most common of them being price correlations [7]. In our work, we test the use of different technical indicators - exponential moving average, relative strength index, average directional index, Bollinger bandwidth - along with the stock price and daily returns to test stock correlations. The final aim of our clustering is that intra-cluster distances are high, while inter-cluster distances are low.

Modified K-Means++ Clustering with PCA:

K-Means++ clustering algorithm, as described by David Arthur and Sergei Vassilvitskii is a popular enhancement over the traditional K-Means clustering algorithm [8]. This technique suggests a unique way to generate the initial centroids for traditional K-Means, in an effort to avoid reaching local optima and quickening the process of convergence. PCA is performed on data primarily for dimensionality reduction, where the individual principal components explain the variance in the data in a descending order. Our data has several features, each as a time series. We reduce this time series data using PCA and use only PC1 as an input for the modified K-Means++ algorithm. For our features (OHLC + 4 technical indicators), PC1 explains 67% of the overall variance in the data. The user inputs two values in the algorithm: the number of stocks needed (N) and a seed stock (seed).

Step 1: Perform PCA on the seed stock, store PC1_{seed}

Step 2: Perform PCA on all other stocks (set I) using seed stock as the fitter

Step 3: Find $\rho_{\text{i}} = corr(PC1_{\text{i}}, PC1_{\text{seed}})$

Step 4: Select stock_{i} such that

$$stock_{\text{i}} = argmin_{\text{i} \in \text{I}} \sum_{j=1}^{J} \rho_{\text{i,j}}$$

where J is the number of already selected stocks (J = 1 for seed stock)

Step 5: seed = stock_{i}, repeat steps 1 – 4 till 'N' stocks selected

Step 4 in the algorithm ensures that we are minimizing the correlations (or maximizing the distance) between the incoming stock and all previously selected stocks. A similar technique is used to choose initial cluster centres in the K-Means++ algorithm, however, we take these N stocks to build our portfolio. Table 1 shows the portfolio built using the algorithm on the training data (2013 - 2017) using N = 10 and seed = Facebook (NYSE: FB).

Iteration	Stock Added	Ticker	Industry
1	Facebook	FB	Information Technology
2	Extra Storage Space	EXR	Real Estate
3	Constellation Brands	STZ	Consumer Staples
4	Fiserv	FISV	Finance
5	OR Auto Parts	ORLY	Industrials
6	Acuity	AYI	Real Estate
7	American Water Works	AWK	Energy
8	Global Payments	GPI	Information Technology
9	Nasdaq	NDAQ	Finance
10	Ecolab	ECL	Energy

Table 1: Stock selction using the modified K-Means++ algorithm

Now that we have built a portfolio (and can build different portfolios using different seed stocks), we evaluate their performance using different metrics. The foremost metric we use is the portfolio Sharpe Ratio, which compares the performance of the portfolio (or any investment) to a risk-free asset, after adjusting for its risk. Sharpe ratio is calculated as:

$$SharpeRatio = \frac{R_p - R_f}{\sigma_p}$$

where R_p is the portfolio return, R_f is the risk free return rate, and σ_p is the standard deviation of the portfolio returns. For all our calculations, we use a risk free rateR_f of 2.5%, based on the average risk free rate from 2013 - 2018. A Sharpe Ratio ≥ 1 is considered good and ≥ 2 is considered very good. Another important metric in portfolio analysis is the Value at Risk (VaR) [9], which measures the gives a confidence interval about the likelihood of exceeding a certain loss threshold. Stated simply, the VaR is a probability-based estimate of the minimum loss in dollar terms expected over a period. VaR is calculated as:

$$\text{VaR } (\%) \ = [R_p - (\text{z-score of interval} \times \sigma_p)] \times 100$$

where we use a 95% confidence interval for all our calculations. The baseline model which we use for comparison is the Markovitz minimum variance portfolio, obtained by solving the following optimization:

$$\text{Minimize } w^T \Sigma w$$

$$\text{subject to } e^T m \geq \mu_d \text{ and } e^T w = 1$$

where $m_{n\times 1}$ is the mean vector, $w_{n\times 1}$ is the weight vector, $e_{n\times 1}$ the vector of ones and $\Sigma_{n\times n}$ is the covariance matrix. We solve this optimization to get the minimum risk weights.

III. Results

The returns given by a portfolio are a strong function of the weights. Optimizing the weights of each security in a portfolio is essential to maximize returns. In order to get the portfolio

with the best Sharpe Ratio and least VaR, we perform a Monte Carlo Simulation of the portfolio risk against return for 10,000 different random weights. Figure 9 shows the resulting plot, with three sets of weights highlighted - a maximum Sharpe Ratio Portfolio, a minimum risk portfolio (from Markovitz Optimization), and a minimum Value at Risk (VaR) portfolio.

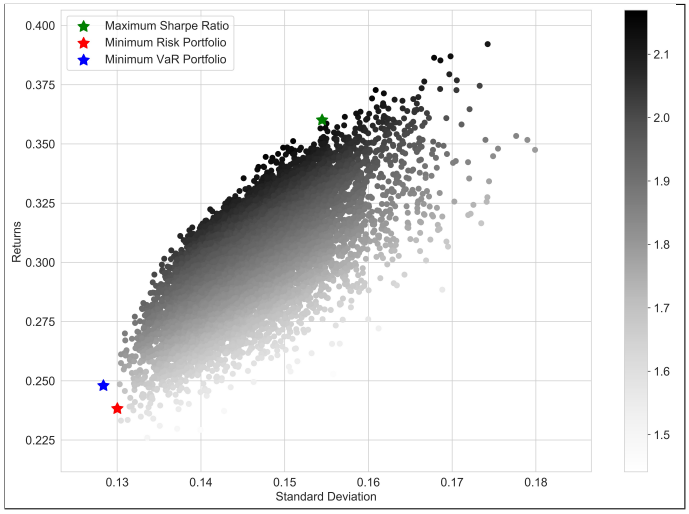

Fig. 9. Risk-return for portfolio (seed: FB) for different weights (Monte Carlo)

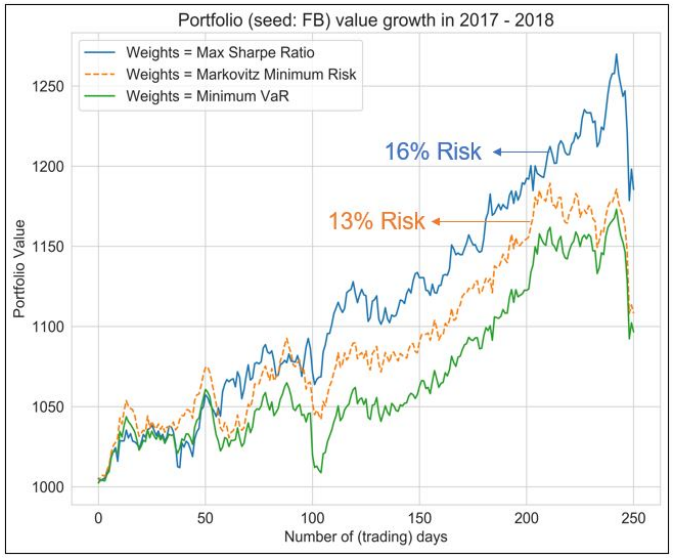

Fig. 10. Value Growth of Portfolios in 2017 - 2018 (seed: FB)

It is important to note that Fig: 9 has been generated using only the training data, viz. the stock data from 2013 - 2017. We want to see how each of the highlighted portfolio behaves through the next year. We, thus, build each portfolio using a $1000 at the start of 2017 and track the value growth over the test data (year 2017 - 2018). Fig: 10 plots each portfolio during 2017 - 2018.

This process of weighing different portfolios using a Monte Carlo plot and testing their value growth in the next fiscal year can be generalized to several other portfolios. We show some good return portfolios in Fig: 11 and Fig: 12 show results for portfolios generated using Nvidia (NYSE: NVDA) and Lockheed Martin (NYSE: LMT) as seed stocks.

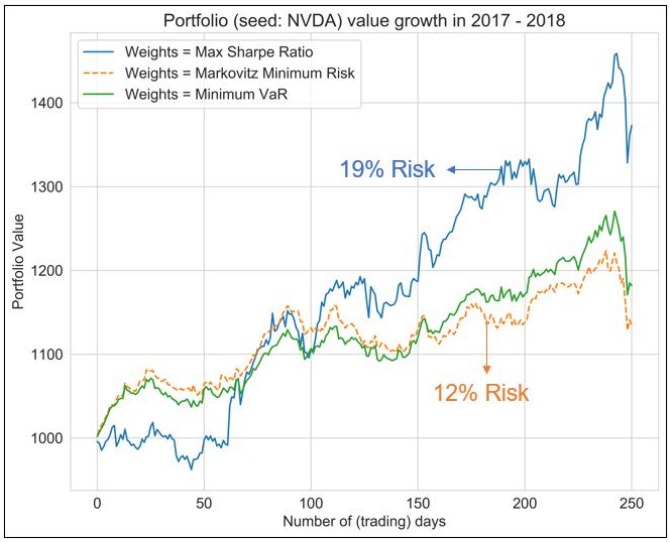

Fig. 11. Value Growth of Portfolios in 2017 - 2018 (seed: NVDA)

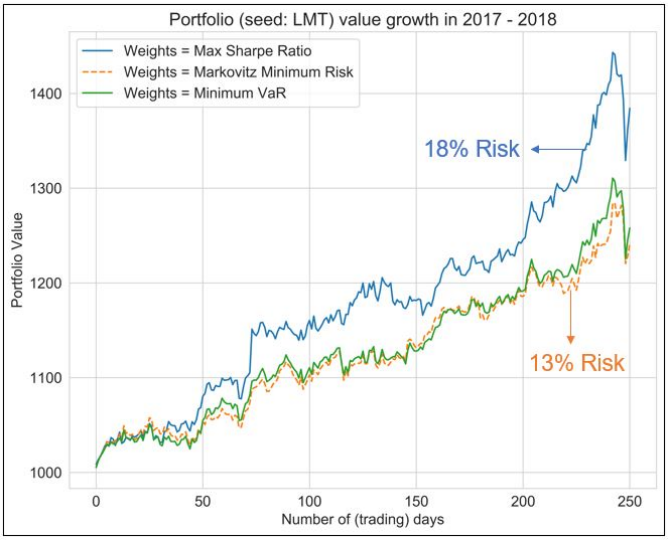

Fig. 12. Value Growth of Portfolios in 2017 - 2018 (seed: LMT)

IV. Discussion

In this project, we worked on constructing optimal equity portfolios using top stocks available in the S&P 500 index. Most of these are large-cap stocks hence risk is relatively low. Hierarchical clustering using SciPy library can only use 2D data as an input. Hence we were able to use only M x N matrix as an input to this algorithm, where M = number of stocks ($\approx$ 500) and N = length of 'close' price time series ($\approx$ 1200). A better algorithm, if developed could use all the OHLCV data, appropriately normalized, to calculate the

distance metric between adjacent clusters. This will provide more qualitative insight into creating a well balanced portfolio. Specifically, systematic risk associated with a market segment (e.g. automotive sector) can be mellowed. If the performance has been bad for a few preceding years and the outlook for the future is not rosy, we can deliberately remove these stocks from our portfolio construction step. Eliminating unsystematic risk which is specific to a single company, is very tough to remove solely on the basis of past data. Qualitative analysis involving founders, balance statements, market competition, trade advantages disadvantages etc. should be taken into consideration if long term investment is in investor's mind.

In our current work on RNNs, we employed past OHLCV data for a single stock to predict how it may perform in the future. Possible future work includes using a more complicated RNN by including data from all stocks to predict one stock. This will have an effect of including complex non linear interactions between multiple different companies and industries. We haven't been able to perform it due the lack of required computational capacity and time.

For the clusters formed using the modified K-means++ algorithm with correlation minimization, we see diverse industries directly being taken into account. Analyzing the three different sets of weights from Fig: 9 - maximum sharpe ratio and minimum risk and VaR - we see that the Max Sharpe Ratio portfolio performs best over the next year (test data). This, however, comes at an additional risk. We would like to emphasize the advantage of creating portfolios using Fig: 11 as an example. The portfolio is created using the Nvidia stock (NYSE: NVDA) as the seed. The portfolio is seen to perform well, which gives over 40% returns over the next year at 19% predicted risk. The Nvidia stock individually performed exceedingly well in the year 2017-2018, giving a 42% return. However, the risk associated with the individual stock was over 45% predicted from previous volatility. By creating a diverse portfolio using Nvidia as the seed stock, we do not greatly compromise on returns over the next year but significantly reduce the expected risk, allowing us to invest with a higher confidence.

To sum up, we emphasize that there is no 'right' portfolio as risk-return depend heavily on the mindset of an investor. Further, a portfolio performing extremely well in one year, due to unforeseen circumstances, may not continue being profitable in coming years. However, by using clustering analysis and minimum correlation stocks, we can reduce the risk of losses to a significant extent while maximizing returns.

V. Member Contributions

We collaborated on most of the aspects of this project. We give here a broad topics on which each individual worked.

- Akhilesh - Feature Generation, Principal Component Analysis, Linear Discriminant Analysis, KMeans Clustering, LSTM-RNN, Presentation and Report.
- Gowtham - Feature Generation, LSTM - RNN, Hierarchical Clustering, Wikipedia Scraper, Portfolio Management Technical Discussions, Presentation and Report.
- Manan - Feature Generation, PCA, KMeans++ Clustering, correlation minimization selection algorithm, Markovitz optimization, Portfolio Evaluation (Sharpe ratio, future returns, risk, VaR, Monte-Carlo weight plots), Presentation and Report.

VI. Acknowledgment

We acknowledge the support and guidance of Professor Ravishankar Iyer at the University of Illinois at Urbana-Champaign. We also thank James Cyriac, Vikram Anjur, Shengkun Cui, and Haotian Chen at the University of Illinois at Urbana-Champaign for their suggestions and feedback.

VII. Disclaimer

Past performance is not indicative of future results. Returns are subject to market risks. We are neither licensed nor qualified to provide investment advice. We will not be liable for any losses that you may suffer arising out of this information. Please do your own due diligence before investing.

VIII. Code Repository

All the project code is available at this GitHub Repository - https://github.com/gowthamkuntumalla/Quant_analysis_stock_market

References

[1] Investopedia, "Were collateralized debt obligations responsible for the financial crisis?" Sep. 2019, https://www.investopedia.com/ask/answers/032315/were-collateralized-debt-obligations-cdo-responsible-2008-financial-crisis.asp.

[2] C. Nugent, "S&p 500 stock data, historical stock data for all current s&p 500 companies," 2018, https://www.kaggle.com/camnugent/sandp500.

[3] S. Exchange, "https://stackoverflow.com/questions/52449331/pca-with-several-time-series-as-features-of-one-instance-with-sklearn."

[4] Wikipedia, "https://en.wikipedia.org/wiki/list_of_s%26p_500_companies."

[5] C. Olah, "Understanding lstm networks," Aug. 2015, https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[6] A. Karpathy, "Understanding lstm networks," May 2015, http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

[7] F. Cai, N.-A. Le-Khac, and T. Kechadi, "Clustering approaches for financial data analysis: a survey," 09 2016.

[8] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding." in *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007, pp. 1027–1035.

[9] D. Harper, "An introduction to value at risk (var)," Jan. 2020, https://www.investopedia.com/articles/04/092904.asp.