Report on

MACHINE LEARNING FOR PORTFOLIO MANAGEMENT

Submitted by:

Lakshmi KS

2K18/MBA/912

Rashika Gupta

2K18/MBA/916

Under the Guidance of:

Mr. Anurag Chaturvedi

Associate Professor

UNIVERSITY SCHOOL OF MANAGEMENT & ENTREPRENEURSHIP

Delhi Technological University

May, 2020

CERTIFICATE

This is to certify that the dissertation report titled "**Machine Learning for Portfolio Management"** is a bonafide work carried out by **Ms. Lakshmi KS & Ms. Rashika Gupta** of **MBA 2018-20** and submitted to University School of Management & Entrepreneurship, Delhi Technological University, Bawana Road, Delhi-42 in partial fulfillment of the requirement for the award of the Degree of Masters of Business Administration.

Signature of Head (USME)

Seal of Head

Place:

Date:

DECLARATION

I, **Lakshmi KS & Rashika Gupta**, student of **MBA 2018-20** of University School of Management & Entrepreneurship, Delhi Technological University, Bawana Road, Delhi – 42, hereby declare that the dissertation report **"Machine Learning for Portfolio Management"** submitted in partial fulfillment of Degree of Masters of Business Administration is the original work conducted by us.

The information and data given in the report is authentic to the best of our knowledge.

This report is not being submitted to any other University, for award of any other Degree, Diploma or Fellowship.

Place: **Lakshmi KS**

Date:

Rashika Gupta

ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards our internal guide, Mr. Anurag Chaturvedi for his support and valuable guidance throughout the duration of the project. We thank him for the constant encouragement and support at every stage.

Our sincere gratitude goes out to our colleagues whose participation in the project gave many valuable inputs for its completion.

ABSTRACT

Artificial Intelligence & Machine Learning has already established a fairly strong foothold in the field of finance & other associated areas. On the front end machine learning is widely utilized for risk management and fraudulent transaction detection. At the front end AI is used for customer segmentation and support and pricing the derivatives (options ,futures & such).

The only arena with limited Machine Learning usage has been in the buy-side of financial activity or more precisely Portfolio Management which includes selection of the best portfolio among a set of portfolios in accordance with certain objective such as expected return, financial risk in short any tangible or intangible aim.

The aim of this project is to review traditional mathematical methods for portfolio optimization (such as Markowitz) , unsupervised (such as Principal Component Analysis), supervised machine learning approaches, and related techniques.

For the purpose of this project we will be using NSE stock data and identifying the top stock options and applying various optimization algorithms using python. Based on the results obtained we plan on identifying the optimal method.

Table of Contents

CHAPTER 1 - INTRODUCTION

Investment and portfolio management is a fairly significant aspect of the secondary capital market due to the fact that it helps in mobilizing the savings of the investors and subsequently assisting in the development of the economy via saving and transfer process. An investor can be a corporation , an individual, a government, or a pension fund. There are a variety of investment options available, offering different risk-reward tradeoffs. A thorough understanding of the core concepts and analysis of the various options available can help the investors in creating a portfolio the maximizes returns while reducing the exposure to risk.

With the availability of a wide variety of investment aveneues , investors have considerable options to build their portfolio after weighing the pros and cons of each option. There are two main categories of investment options i.e.,

- Financial assets : equity shares, derivative instruments , government securities, post office schemes, mutual fund shares, corporate debentures ,insurance policies, and deposit with banks.
- Real assets : tangible assets such as house, gold, commercial property, agricultural farm, precious stones, and art objects.

There are two broad categories of investing – direct and indirect investing . Direct investing is where investors manage their own individual portfolios and the risk and return they receive is solely dependent on their ability to analyze the market behaviour and fluctuations. Indirect investing involves financial intermediaries which invest pools of funds into the market and maintain the investor portfolios, providing the investors with expert advice and recommendations and relieving the investors of making their own decisions.

This is the point where portfolio managers come in, their primary job is to manage investor portfolios framed according to their invidual preferences ,return objectives and risk bearing capacity. Portfolio management entails portfolio planning, identification, selection and construction, feedback and evaluation of securities. The hidden talent in portfolio management lies in obtaining an adequate balance between the objectives of safety, liquidity and profitability.

Traditionally after setting up the investment policy and portfolio objectives by assessing the current and future financial needs of the individual investor, the next task for the investor/portfolio manager is the analysis and evaluation of the investment options performed with the help of a combination of technical and fundamental analysis. Technical analysis aims to predict the future movement of the price of a particular financial asset that is traded on the market , one of the main drawback of technical analysis is the fact that the analysis of historical prices is based on the assumption that trends and patterns repeat itself.

On the other hand fundamental analysis aims to determne the intrinsic value of an particular financial asset, it helps in determining which financial asset is over-priced or under- priced based on the difference between their market value and intrinsic value. Similar to technical analysis , fundamental analysis is also prone to errors and bias one of the assumptions that gives rise to the same is the fact that fundamental analysis assumes that intrinsic value is the present value of future flows from particular investment.

Thus there was felt a need to provide sound and accurate analysis of investment instruments in order to construct even more risk - resistant and high return yielding portfolios. Modern portfolio theory (MPT) gave rise to the same. MPT or portfolio theory was first introduced by Harry Markowitz in his paper “Portfolio Selection” in the Journal of Finance (1952). Before modern portfolio theory (MPT), the decision about whether to include a investment option in a portfolio was based solely upon the fundamental analysis of the firm, its dividend policy and its financial statements. Harry Markowitz stirred a whirlwing by suggesting the fact that the value of a security to an investor might can be evaluated optimally by calculating its mean, its standard deviation and its correlation to other securities in the portfolio.

Portfolio theory evaluates how risk - averse investors frame portfolios in order to optimize expected returns for a given level of market risk. The theory also aims to quantify the benefits of diversification. Portfolio theory constructs an efficient frontier of optimal portfolios out of a universe of risky assets . Every portfolio on the efficient frontier provides the maximum expected return for a particular level of risk. Investors are required to hold one of the optimal portfolios on the efficient frontier and adjust their total market risk positions within the risk - free financial asset.

Furthermore in this study we have sought to explore the applications of Machine Learning algorithms in the field of investment and portfolio management Machine learning based methods that refer to statistical learning with data are widely applicable in computational finance.Which is found to be particularly helpful in order to obtain more accurate and risk resistant portfolios and also to overcome the limitations and shortcomings of the traditional portfolio optimization techiques. Some of the Machine Learning lagorithms that we have tried to use are Principal Component Analysis (PCA), Auto-encoder risk, Hierarchical Clustering, etc.

1.1 Industry Profile

1.1.1 Investment options

There is wide variety of investment vehicles/options now available to investors and portfolio managers. Investors are free to select any one or more alternative options depending on their needs. All categories of investors are equally interested in safety, liquidity and reasonable return on the alternatives invested by them. In India, investment alternatives are continuously increasing along with new developments in the financial market. For sensible investing, investors should be aware of the the characteristics and features of various investment options.

- Equity Shares : type of security that represents ownership in a company. Investment in shares is more risky because the share prices go on changing day by day. Today, the market is more volatile and hence fluctuating share leading to lack of stability. However , returns on equities over a long time horizon are generally higher than most other investment options.

- Bonds & Debentures : are debt instrument issued for a period of more than one year with the purpose of raising capital by borrowing. Debenture is a document issued by a company while bond is issued by the Government. There is not much risk while e investing in debentures as compared to shares. The return on debentures is also reasonable and stable.

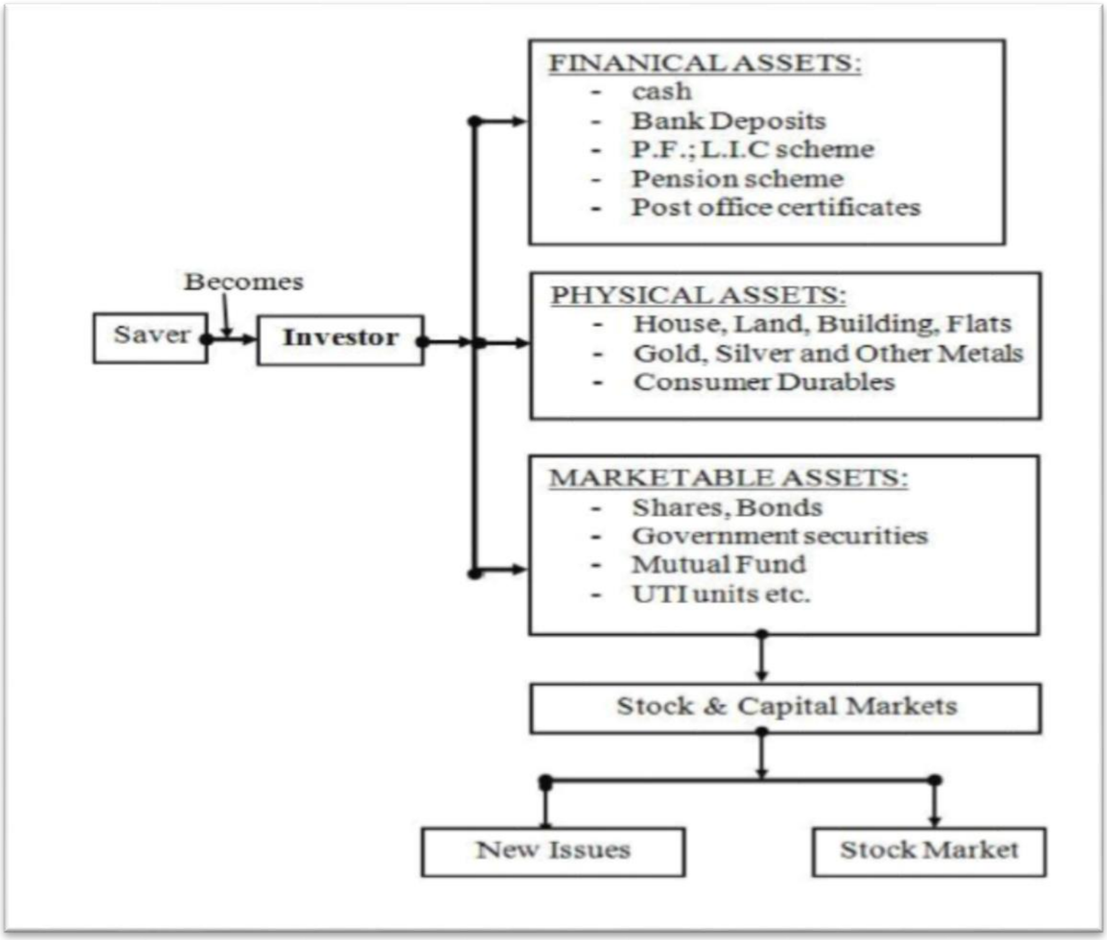

Fig 1.1 Investment Options

- Mutual Funds : is made up of money that is pooled together by a large number of investors. The investments by the Mutual Funds are made in shares, bonds, debentures,etc. The current value of Mutual Funds is calculated on daily basis and is reflected in Net Asset Value (NAV) declared regularly and NAV keeps on fluctuating with the changes in the stock and bond market. Hence the investing in

Mutual Funds is not risk free, but a good fund manager can assist in providing regular and higher returns compared to the returns from fixed deposits of a bank. Mutual Funds provide liquidity in case of open-ended schemes and some of the schemes provide tax-relaxation.

- Life Insurance Policy : The life insurance sector in India has been gradually developing at a steady pace since the last few decades with a lot of new private players entering the market. Life insurance is a kind of investment option that serves a dual purpose - provides family protection to the investor as well as return on investment in the form of yearly bonus on the policy. The return on investment is reasonably low 6% because of risk exposure and tax relaxation. The premium paid on a life insurance policy is exempted from the taxable income . Even though the maturity period of a life insurance policy is long, it can be liquidated or loan can be availed on the policy, hence there is some level of liquidity in this investment avenue. Therefore, life insurance is a profitable investing option and there is almost negligible risk in it.

- Bank Deposits : Investing of surplus money in bank is quite popular in India especially among service class population. Deposits are made by the account holders for specific period of time and the bank pays interest on it. Bank deposits have high level of liquidity. Banks also provide loans on the security of fixed deposit receipts. One of the main limitations of bank deposits as an investment option is that the rate of return is low when compared to other investment options. Also capital appreciation is not possible when investing in a bank.

- Investment in Real Estate : includes properties like industrial land, building, agricultural land ,plantations, farm houses , flats or houses. As the demand increases but the supply of land is limited, the prices tend to increase. Hence, it is lucrative investment option which provides higher return within a short period of time. But in this option there is a low liquidity and the risk in this type of investment is higher when compared to investment in banks and mutual funds.

- Investment in Silver & Gold : in India the value of precious metals is deeply rooted in the culture and customs, with gold being used as most sought after wedding gift since centuries. The prices of gold and silver are rising at rapid rate, because of the fact that the supply is increasing at a lower rate than the demand which is quite high. This investment option is highly liquid, as it could be sold at any time. The market prices are continuously climbing. Hence, the return on investment avenue is also increasing. The investment is also safe and secured combined with the advantage of capital appreciation.

1.1.2 Objectives of Portfolio Management

Any portfolio management decision will be influenced by three objectives security, liquidity and yield. A best investment decision will be one, which has the best possible compromise between these three objectives :

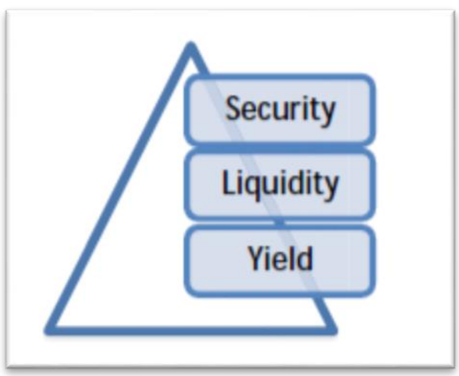

Fig 1.2 Objectives of Portfolio Management

- Security/Safety of Principal: Security not only involves keeping the principal sum intact but also keeping intact its purchasing power intact. Safety means protection for investment against loss under reasonably variations. In order to provide safety, a careful review of economic and industry trends is necessary. In other words errors in portfolio are unavoidable and it requires extensive diversification. Every investor wants his basic amount of investment should remain safe.

- Liquidity i.e. nearness to money: Because investors may need to convert their investment back to cash or funds to meet their unexpected needs and demands , their investment should be highly liquid. They should been cashable at short notice, without any difficulty and minimal loss . If they cannot come to our rescue, investors may have to borrow or raise funds externally at high cost and at unfavorable terms and conditions. Such liquidity can be possible only in the case of investment, which has always-ready market and willing buyers and sellers. Such instruments of investment are called highly liquid investments.

- Diversification: The basic objective of building a portfolio is to reduce risk of loss of capital and / or income by investing in various types of securities and over a wide range of industries.

- Yield : best described as the net return out of any investment. Hence given the level or kind of security and liquidity of the investment, the appropriate yield should encourage the investor to go for the investment. If the yield is low compared to the expectation of the investor, they might prefer to avoid such investment and keep the funds in the bank account or in worst case, in cash form in lockers. Hence yield is the attraction for any investment and normally deciding the right yield is the key to any investment for any portfolio manager.
- Stability of Income: So as to facilitate planning more accurately and systematically the reinvestment consumption of income is important.

- Capital Growth: This can be attained by reinvesting in growth securities or through purchase of growth securities. Capital appreciation has become an important investment principle. Investors seek growth stocks which provides a very large capital appreciation by way of rights, bonus and appreciation in the market price of a share.

- Marketability: It is the case with which a security can be bought or sold. This is essential for providing flexibility to investment portfolio.

- Favorable Tax status (Tax Incentives): The effective yield an investor gets via his investment depends on tax to which it is subjected to . By minimizing the tax burden, the yield can be effectively improved. Investors try to minimise their tax liabilities from the investments. The portfolio manager has to keep a list of such investment vehicles along with the return risk, profile, tax implications, yields and other returns. Investment programs without considering their subsequent tax implications may prove costly to the investor.

1.1.3 Portfolio Management Industry- Return

A return is the ultimate objective for any investor and every type of investment avenue can be characterized by certain level of profitability. But a adequate balance between return and risk is a key concept of investment and portfolio management. The most basic definition of return is the benefit reaped from an investment. In most cases the investor can estimate his/ her historical return precisely.

Most investments have two components of their measurable return - some form of income and a capital gain or loss. The rate of return is the percentage increase in returns with respect the holding period :

$$Rate\ of\ return = \frac{Income + Capital\ gains}{Purchase\ price (\%)} \ldots\ldots\ldots\ldots (1.1)$$

For example, rate of return of the share (r) will be estimated as :

$$R = D + \frac{Pme - Pmb}{Pmb} (\%) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.2)$$

Here   D – dividends;

P_{mb}- market price of stock at the beginning of holding period;
P_{me} - market price of stock at the end of the holding period;

The rate of return, calculated in equations 1.1 and 1.2 is called holding period return, due to the fact that its calculation is independent of the variable of time. The investor only knows about the beginning of the investment period and its end. The investor has to be very careful during the interpretation of holding period returns in portfolio analysis as it is impossible compare the alternative investments using holding period returns, if their holding periods have been different. Statistical data that can be used for the investment and portfolio formation analysis deals with a series of holding period returns. In such cases arithmetic average return or sample mean of the returns (ř) can be calculated as :

$$\check{r} = \frac{\Sigma ri}{n} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1.3)$$

Here r_i - rate of return in period i;
n - number of observations ;

It must be kept in mind that both holding period returns and sample mean of returns have been calculated using historical data. Howsoever , what happened in the past for the investor is not as important as whatever happens in the future, because all the decisions of the investor are focused on the expected results from the investments which occur in the future.

In order to analyze each particular investment option's probability to earn profit in the future , the investor must think about several scenarios of possible fluctuations in macro economy, industry and the organization which could influence asset prices ant rate of return.

Hypothetically , it could be a series of discrete possible rates of return in the future for the same asset with the different probabilities of earning the particular rate of return. But for the same asset the sum of all probabilities of these rates of returns must be equal to 1 or 100 % , called simple probability distribution in mathematics.The expected rate of return E(r) of investment is defined as the statistical measure of return, which includes the sum of all possible rates of returns for the same investment weighted by their respective probabilities :

$$E(r) = \Sigma hi * ri \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.4)$$

Here h_i - probability of rate of return;

r_i - rate of return.

Cases where the investor has more than enough information for modeling of future scenarios of changes in rate of return for investment, the decisions should be based on estimated expected rate of return. But sometimes sample mean of return (arithmetic average return) are a useful proxy for the concept of expected rate of return. Sample mean can give an unbiased estimate of the expected value, but obviously it's not perfectly accurate, because based on the assumption that the returns in the future will be the same as they were in the past. In general, the sample mean of returns should be taken for the cases of longer duration, as the investor is positive that there has not been too much change in the shape of historical rate of return probability distribution.

1.1.4 Portfolio Management Industry- Risk

The most important characteristics of investment avenues on which the foundation of the overall variety of portfolios is based upon is the return on investment and the risk which is defined as the uncertainty of the actual return that can be earned on that investment. A portfolio manager has to explore ways in which the risk can be minimized with respect to a desired level of return on the investment or maximize the return according to the constraints of a particular level of risk.

There are mainly two broad categories of risk – Systematic & Unsystematic risk.

Systematic risk also referred to as non-diversifiable risk or market risk. It is defined as the fluctuations in the returns on securities that occur due to macroeconomic factors. These factors can be the political, social or economic , that affect the business. Systematic risk can be caused due to unfavorable reasons such as an act of nature like a natural disaster, changes in government policy, international economic components, etc. Systematic risk distresses a large number of organizations in the market or an entire industry sector. It can be eradicated through several ways like asset allocation or hedging.

The Great Recession of 2008 proves to be a prime example of systematic risk. People who had invested in all types of securities saw the values of their investments plummet down due to the market-wide economic event. The great recession affected various securities in diverse ways. Hence investors who held stocks were affected in worse ways as compared to those with wider asset allocations. Systematic risk is further divided into three categories - Interest, Inflation and Market risk.

Unsystematic risk is defind as the fluctuations in profits of a company occuring due to micro-economic factors . These risk factors exist within the company and could be avoided if necessary action is taken. The risk factors can include the production of undesirable products, labor strikes, etc. Unsystematic risk is caused by internal factors and can be controlled and avoided, up to some extent via portfolio diversification. Unsystematic risks can be further divided into – Business and Financial risks.

After obtaining an acceptable level of risk for an investor's portfolio by analyzing their individual time horizon and bankroll, portfolio managers can use the risk pyramid approach to balance their investment options. This pyramid can be thought of as an asset allocation tool that investors can use to diversify their portfolio investments according to the risk profile of each security. The pyramid, representing the investor's portfolio, has three tiers :

Base of the Pyramid is the foundation of the pyramid representing the strongest portion. This area should be composed of investment optiond which are low in risk and have tangible expected returns. It is the largest area and composes the bulk of your assets.

Middle Portion is the area made up of medium-risk investments that provide a relatively stable return while still giving room for capital appreciation. Although more risky than the investment avenues creating the base, these investments can still be considered relatively safe.

Summit is the composed of specifically high-risk investments, it is the smallest area of the portfolio. The amount of funds invested in the summit should be quite disposable so that the investor doesn't have to sell prematurely in situations of capital losses.

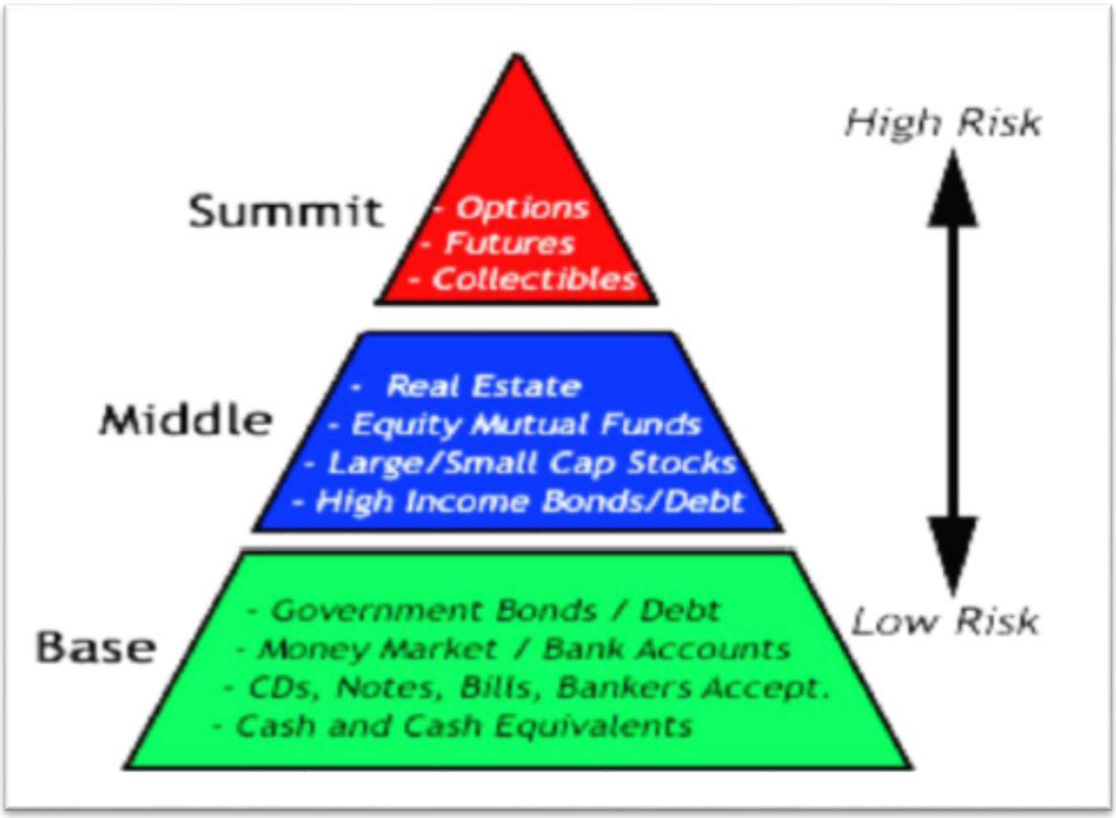

Fig 1.3 Investment Risk Pyramid

Investing in securities pre-supposes risk. One of the most common way of reducing risk is to follow the principle of diversification. In a diversified portfolio, some investment avenues might not perform as expected but others might exceed expectations leading to the effect that the actual results of the portfolio would be reasonably nearer to the desired results.

Risk is associated with the dispersion in the likely outcome. And dispersion entails variability. Hence, the total risk of investment options can be measured via – Variance and Standard Deviation.Variance can be calculated as the deviation of each possible investment's rate of return from the expected rate of return :

$$\delta^2(r) = \sum_{i=1}^{n} h_i * [r_i - E(r)]^2 \ldots\ldots\ldots\ldots\ldots(1.5)$$

In order to compute the variance in equation (1.5) all the rates of returns which were observed in estimating expected rate of return (r_i) have to be taken along with their probabilities of occurance (h_i).

Another equivalent to obtain the total risk is standard deviation which is calculated as the square root of the variance :

$$\delta(r) = \sqrt{\sum_{i=1}^{n} h_i * [r_i - E(r)]^2} \ldots\ldots\ldots\ldots\ldots(1.6)$$

When the arithmetic average return or sample mean of the returns (ř) is used instead of expected rate of return, sample variance ($\delta^2(r)$) can be calculated as :

$$\delta^2(r) = \frac{\sum_{t=1}^{n}(r_t - ř)^2}{n-1} \ldots\ldots\ldots\ldots\ldots(1.7)$$

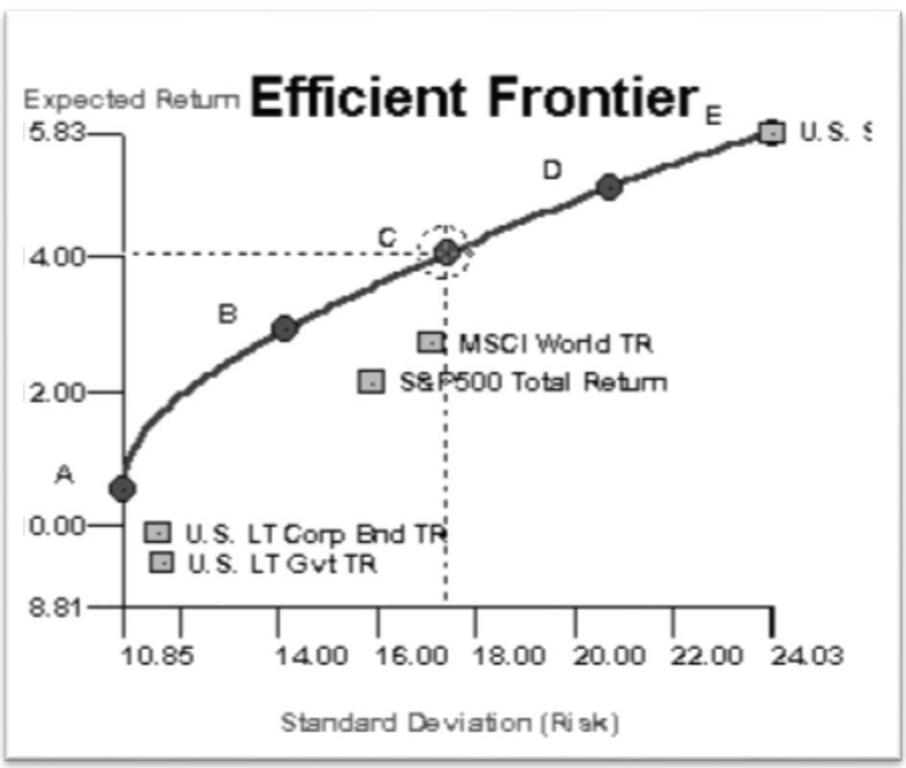

Fig 1.4 Graph of Expected Returrn vs Risk

Similarly Sample standard deviation (δ_r) can be calculated as the square root of the sample variance :

$$\delta_r = \sqrt{\delta_r^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(1.8)$$

Variance and the standard deviation are similar measures of risk which could be used for the similar purposes in investment analysis however, standard deviation is used more commonly. Variance and standard deviation are used when investor is trying to estimate the total risk that is expected in the particular period in thenear future. Sample variance and standard deviation are more commonly used in the cases when investor evaluates total risk of their investments during historical period.

There are several types of risk associated with investment options and so are a number of returns and profits associated with investment avenues but the maintaining of the balance between risk and return is of the prime imoortance for any portfolio manager. There are several techniques and approaches that quantify the level of risk and returns associated with various types of investment options and the preferences of the investors, but in the end the optimization of the profitability and risk exposure can only be kept in check through thorough investigation and analysis of investment avenues and diversification of portfolios in order to mitigate the losses incurred in one investment option via the profit earned in order investment instruments.

1.1.5 Types of Portfolio Optimization

It is a widely held consensus that portfolio optimization is a powerful tool that can be used as a mechanism to make best use of all the information available to investors. While portfolio optimization has a strong theoretical merit, it is not particularly useful in practice. Portfolio managers are concerned that optimization is an error maximizing process entailing numerous estimation issues.

The Modern Portfolio Theory (MPT) which was heavily influenced by the principles of Harry Markowitz who came up with the use of mean-variance for asset valuation posed some major problems when used practically because of the fact that mean-variance optimization has a tendency to maximize the effects of errors in the input assumptions.

The methods of portfolio construction and optimization primarily takes the following approaches –

- Market capitalization weight - investments are held in accordance to their market capitalizations. The Capital Asset Pricing Model (CAPM) described by Sharpe and Treynor proposed a linear relationship between returns and an investment's β coefficient (the sensitivity of the asset returns to market returns). CAPM suggested that the market rewards the investors for the risk of being invested in the market , and not for any stock-specific risks that can be diversified away. Investors who believe that the CAPM expresses a justifiable relationship between risk and return will select to hold the market portfolio. CAPM is a model for pricing an individual security or a portfolio and enables to calculate the reward-to-risk ratio for any security in relation to the overall market. Theoretically, a security is correctly priced when the observed price is equal to the value calculated using the CAPM discount rate. If the observed price is higher than the valuation, then it can be said taht the security is overvalued. But one of the main limitation of CAPM is that it suffers from poor explanatory power in practical use due to the fact that returns have historically exhibited only a weak relationship with market β coefficient.

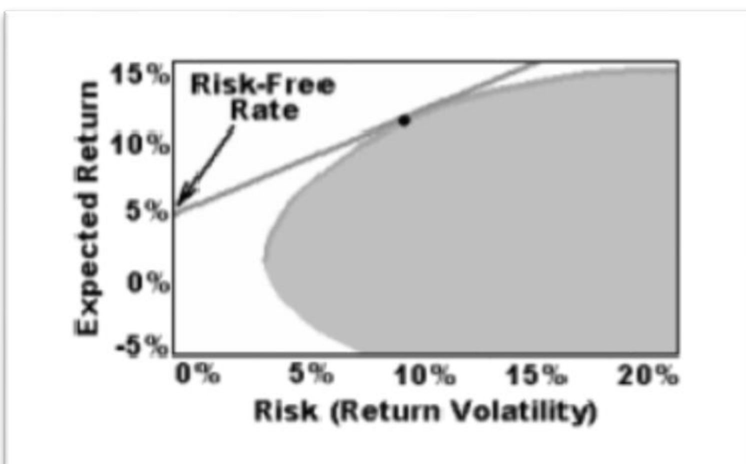

Fig 1.4 Capital Market Line

- Minimum Variance based optimization - The primary objective of the Minimum Variance portfolio is that if all investments have equal expected return independent of risk factor, investors desiring maximum returns for minimum risk should concentrate exclusively on minimizing risk. It was assumed that relationship between risk and return should be ignored, at least in the case of equities

$$\omega^{MV} = \arg min.\, \omega^{T}.\Sigma\omega \ldots\ldots\ldots\ldots\ldots..(1.9)$$

Here – Σ is the Covariance matrix ;

Due the presence of a faulty link between risk and return,it was established that regularly reconstituted lonng-buy only Minimum Variance portfolio could compensate for the capitalization weighted portfolio for stocks.

- Maximum Diversification based optimization - in this approach it is assumed that the markets are risk-efficient, in a way so that the investments would produce returns in accordance to total risk, measured by volatility. This differs from CAPM, which assumes returns are in accordance to non-diversifiable (i.e. systematic) risk. In accordancewith the view that returns are directly proportional to volatility, the Maximum Diversification optimization replaces asset volatilities for returns in a maximum Sharpe ratio optimization, defined by :

$$\omega^{MD} = \arg max \frac{\omega * \sigma}{\sqrt{\omega^{T} \quad + \Sigma\omega}} \ldots\ldots\ldots\ldots\ldots\ldots(1.10)$$

Here σ - is reference a vector of volatility;

Σ - is reference a vector of the covariance matrix ;

It can be clearly concluded that volatility of a portfolio of positively correlated investments could be equal to the weighted sum of the volatilities of their components, because of the fact that there can be no opportunity for diversification. While in the case where the assets are imperfectly correlated, the weighted average volatility becomes larger than the portfolio volatility in accordance to the available level of diversification.

- Maximum Decorrelation based optimization – this approach is very closely related to the aforementioned Minimum Variance and Maximum Diversification, but the only differing point is that it is only used in the cases where an investor believes that all assets have similar returns and volatility, but heterogeneous correlations. It is defined as a Minimum Variance optimization performed on the correlation matrix instead of the covariance matrix. Maximum Decorrelation portfolio can be obtained via :

$$\omega^{M\ Dec} = \arg\min \omega^{T} \, . A \, . \, \omega \quad \ldots\ldots\ldots\ldots\ldots(1.11)$$

  Here A – is the correlation matrix ;

  One major limitation of this approach is that Maximum Diversification portfolio tends to concentrate at assets with high volatility and low covariance with respect to the market.

- Inverse Volatality & Variance based optimization – aims to overcome the drawback of Minimum Variance & Maximum Diversification that they can be fairly concentrated when the number of assets is small. Concentrated portfolio might not be able to account for large amounts of capital without high market impact costs. In addition, concentrated portfolios tend to be more susceptible towards mis estimation of volatility.

  When investments have similar Sharpe ratios and an investor can't adequately estimate correlations, the optimal portfolio will be weighted in proportion to inverse of the assets' volatility. In case of investments having similar expected returns (irrespective of volatility) and anonymous correlations, the Inverse Variance portfolio is mean-variance optimal. The weights for the Inverse Volatility and Inverse Variance portfolios are defined via :

$$\omega^{IV} = \frac{1/\sigma}{\sum_{i=1}^{n} 1/\sigma} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1.12)$$

$$\omega^{IV\,ar} = \frac{1/\sigma^2}{\sum_{i=1}^{n} 1/\sigma^2} \quad \ldots\ldots\ldots(1.13)$$

Here σ - is the vector of asset volatilities ;

σ^2 – is the vector of asset variances ;

1.1.6 Government framework & guidelines

Securities and Exchange Board of India (SEBI) was established in 1988 with the prime objective to provide protecion to the investors and their vested interests in the capital market and also to restore the faith of investors and other constituents of the market. It was also required to promote the development of capital market and to regulate the securities market. The trust of the investors and their confidence are the key requirement for the development and overhaul of a flourishing and smooth capital market. The role of regulator in establishing and enhancing investor confidence is hence forth of the prime importance.

The prime focus of SEBI is on investor protection which it ensures via

- Penalties which can be imposed by on different intermediaries, for failures/defaults, a failure to furnish information and maintainance of books of accounts, failure to enter into an agreement with clients/to redress investors grievances; failure by an AMC to observe rules/regulations; defaults in case of mutual funds/stock brokers; insider tradings; non-disclosure of substantial acquisition of shares and takeover bids; fraudulent and unfair trade practices.
- Ombudsman Regulation – In order to redress the grievances of investors, SEBI established an ombudsman/stipendary ombudsman. An ombudsman is an official who would receive complaints and assist in providing their resolution by amicable settlement, approve an amicable settlement between parties and adjudicate complaints in the event of failure of a settlement.

- SEBI Intermediary Regulation - The main components of the regulation of intermediary entails registration, obligations, inspection and disciplinary proceedings and action in case of default of all intermediaries associated with the securities market such as brokers/sub-brokers/bankers to an issue, registrar to an issue, trustees, underwriters, merchant bankers,portfolio managers, investment advisor c,redit rating agencies and other intermediaries associated with the securities market in any manner that is laid down by SEBI.

Portfolio management service is among the merchant banking activities acknowledged by Securities and Exchange Board of India (SEBI). Portfolio management service could either be rendered by the SEBI recognized category I and category II merchant bankers or portfolio managers or discretionary portfolio manager as it is defined in clause (e) and (f) of rule 2 of SEBI (portfolio managers) Rules ,1993. Recognizing the importance of portfolio management services, the SEBI has also defined certain specific guidelines for the proper and professional conduct of portfolio management services. As per the aforementioned guidelines only recognized merchant bankers registered with SEBI are authorized to offer those services. A portfolio manager by virtue of his knowledge, background and experience is expected to study the various avenues available for profitable investment and advise his client to enable the latter to maximize the return on his investment and at the same time safeguard the funds invested.

1.2 Organization Profile

Portfolio management has been one among the prime components of the secondary capital market. Besides serving the aim of mibilizing the savings of the masses and pooling them for investment purposes so as to produce assistance within the economic stimuli of the country , it also helps in maintaining the balance of price formation and maintaining liquidity equilibrium within the market successively helping in establishing the method of capital formation for organizations.

Portfolio optimization is kind of a frequently studied problem in fields of mathematics, finance , statistics furthermore as because the growing area of knowledge analytics. In theory portfolio optimization is primarily related to determining the weights reminiscent of each investment option. But in real because of increasing market fluctuations ,price volatality, business constraints and individual investor preferences requires the urgent need of an adequate computational facility so as to judge and analyze the many complex and ever increasing parameters.

Traditionally portfolio managers relied and other players during this arena mostly relied on their ability and intution by performing the basic analysis of the investment options like shares, bonds, debentures and mutual funds via doing background check on the organization issuing them its financial state, financial statements like record and income statement and its performance on the stock exchange. But gradually with the technological advancement within the field of knowledge analytics especially within the field of knowledge mining and massive data analytics has opened a completely new arena for finance with a solid foundation and supply adequate capabilities to analyse large datasets with multiple parameters , overcoming the limitation of traditional statistical models which fail to produce reliable leads to the presence of enormous datasets.

1.3 Objective of the Study

The prime goal of this study is to identify potentially optimum technique that can be utilized for portfolio management and subsequent portfolio optimization by comparing and evaluating both traditional statistical portfolio optimization techniques as well as supervised and unsupervised Machine Learning algorithms against the market benchmark. The study presented is exploratory in nature attempting to spot and recognize the optimum approach for portfolio management starting from traditional ones like Markowitz to Machine Learning algorithms.

We have utilized four variations of Markowitz, then use PCA , Autoencoder ,HRP and Smoothening. The main hypothesis for this study has been :

H_0 : There is no difference in the return ,volatality, α , β and Sharpe ratio calculated for these methods.

H_1 : Difference is found in the return ,volatality, α , β and Sharpe ratio calculated for these methods.

The objective is further divided into the following stages :

- Stage I :
  Data cleaning & mapping of prices to stocks.
- Stage II :
  Smoothening of the data
- Stage III :
  Setting up the objectives of your portfolio market benchmark (β coefficient) and the Sharpe Ratio .
- Stage IV :
  Applying unsupervised ML algorithm Principal Component Analysis in order to obtain eigenportfolios
- Stage V :
  Applying Auto-encoder to obtain hierarchy and overcome the limitations of PCA.
- Stage VI :
  Applying the supervised learning model.
- Stage VII :
  Comparing the results of all the techniques with the market benchmark and evaluating the best possible fit.

CHAPTER 2 – LITERATURE REVIEW

Portfolio management is one of the significant aspect of primary and secondary capital markets. Essentially portfolio management involves the identification & subsequent selection of a group of financial assets such as debt instruments, shares, stocks, bonds, cash equivalents, mutual funds, etc. In order to avoid less than optimum yield portfolio management is governed scientifically by four factors - Risk, Return, Safety and Liquidity. Of the aforementioned factors Risk and Return are of prime importance, with the objective of portfolio management being to maximize return and minimize risk of the non-performing financial assets.

Duly certified portfolio managers act on behalf of clients in accordance with a variety of approaches to portfolio management such as – active , passive portfolio management and discretionary , non-discretionary portfolio management with varying levels of control.

Similar to any major activity in Indian capital markets, portfolio managers are also liable to the guidelines of SEBI (Securities & Exchange Board of India). The first set of guidelines was issued in the year 1993 after which due to growing market demand and need for more diverse and customized investment portfolios required a complete overhaul in the form of the SEBI (Portfolio Managers) Regulations, 2020. The new regulations sought out to bring about greater transparency in a largely unregulated field to promote healthier competition and ethical transactions ultimately to keep fraudulent activities in check and maintain the attractiveness & viability of the industry.

Modelling the behavior of financial markets has remained a challenge for a long period of time due to the volatility, uncertainty and lack of stationarity all resulting in poor predictive quality when using traditional mathematical models. But since the advent of technology especially in the fields of Artificial Intelligence and Machine Learning we now have the opportunity to obtain a more efficient asset selection for a gamut of clients with varying investment needs and objectives with a help of supervised and unsupervised machine learning algorithms such as Clustering, PCA (Principal Component Analysis) and other advanced techniques such as reinforcement learning.

M. Ivanova and L. Dospatliev (2017) realized that portfolio optimization is a crucial factor in determining the strategies for investors both retail and institutional. The goal of portfolio optimization is to maximize portfolio return and minimize portfolio risk. As the return varies based on risk stockholders have to trade-off the inconsistency between risk and return for their savings. Therefore, there is no one particular optimized portfolio that satisfies all investors. The optimal portfolio is calculated by the inclinations of the saver's risk and return.

The objective of M. Ivanova and L. Dospatliev's report was to determine on the efficient limits in summing up the three optimal portfolios (Minimum risk portfolio, Maximum portfolio return for a given level of risk, and Maximum Sharpe ratio portfolio) using Markowitz Theory. The data inputs for this study was weekly closing prices of 50 stocks traded on Bulgarian Stock Exchange between January 2013 and December 2016.

In the study as of December 2016, there are 516 corporations listed on the Bulgarian Stock Exchange. Fifty of them were chosen for the study based on two criteria:

- top50th largest market capital on the stock market;

- the listed companies have minimum 4 years of listing on the market from January 2013

The study used Portfolio Efficient Frontier and Sharpe Ratio. They compute the average weekly return for each of the 50 assets. There are 47 out of 50 sample stocks yielding profitable returns in the study period. The remaining 6% had poor performance with negative rate of return. Risk Return was drawn and using Markowitz model, minimum risk portfolios were obtained.

Every possible asset combination can be plotted in risk-return space, and the collection of all such possible portfolios defines a region in this space. The line along the upper edge of this region is known as the efficient frontier sometimes called the" Markowitz bullet". Combinations along this line represent portfolios (explicitly excluding the risk-free alternative) for which there is lowest risk for a given level of return. Conversely, for a given amount of risk, the portfolio lying on the efficient frontier represents the combination offering the best possible return. Mathematically the efficient frontier is the intersection of the set of portfolios with minimum risk and the set of portfolios with maximum return.

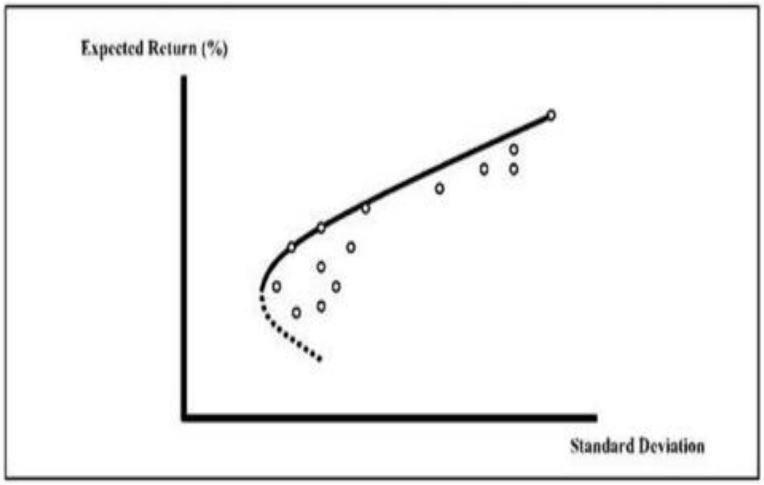

Fig 2.1 Sharpe Ratio

Sharpe’s measure is also known as Sharpe ratio or reward-to-volatility ratio. Sharpe’s measure is a measure of portfolio performance that gives the risk premium per unit of total risk, which is measured by the portfolio’s standard deviation of return. The risk premium on a portfolio itself is the total portfolio return minus the risk-free rate. In other words, Sharpe’s measure divides average portfolio excess return by the standard deviation of returns on the same time period. Sharpe’s measure can be expressed in the following formula

$$S_P = \frac{E(r_P - r_f)}{\sigma_P}, \qquad (2.1)$$

The research attained its goal of the practical application of Markowitz model to set up an optimal portfolio included stocks traded in Bulgarian stock market during the said. It was noted that during the study time period the portfolios formed by Markowitz model performed better than any domestic individual security. By investing in efficient portfolios - the ones located on the efficient frontier, investors afford to get maximum return on investment given a certain level of risk, maximum Sharpe ratio, or a minimum risk. It is the power of Markowitz diversification by seriously taking into account covariance and correlation between assets. Accordingly, Bulgarian investors, if knowing how to properly apply Markowitz model, certainly can improve their investment performance.

Ye Wang, Yanju Chen, and YanKui Liu (2016) in their paper study the portfolio selection problem using hybrid decision systems. Firstly, the return rates are characterized by random fuzzy variables. The goal is to maximize the total expected return rate. For a random fuzzy variable, this paper defines a new equilibrium risk value (ERV) with credibility level beta and probability level alpha. As a result, our portfolio problem is built as a new random fuzzy expected value (EV) model subject to ERV constraint, which is referred to as EV-ERV model.

Under mild assumptions, the proposed EV-ERV model is a convex programming problem. Furthermore, when the possibility distributions are triangular, trapezoidal, and normal, the EV-ERV model can be transformed into its equivalent deterministic convex programming models, which can be solved by general purpose optimization software. To demonstrate the effectiveness of the proposed equilibrium optimization method, some numerical experiments are conducted. The computational results and comparison study demonstrate that the developed equilibrium optimization method is effective to model portfolio selection optimization problem with twofold uncertain return rates.

When the randomness of uncertain return rates follows normal distributions, the proposed equilibrium portfolio selection model was turned into an equivalent credibilistic portfolio optimization model. The convexity of the credibilistic portfolio optimization model was discussed, which facilitates finding the desired global optimal portfolio.

Furthermore, when the fuzziness of uncertain return rates follows trapezoidal, triangular, and normal distributions, the credibilistic portfolio optimization model was turned into its equivalent deterministic convex programming models.

Ye Wang, Yanju Chen, and YanKui Liu compared the proposed equilibrium optimization method with traditional stochastic optimization method via a portfolio selection problem. The computational results demonstrated that both optimization methods can provide diversified investment schemes. However, the obtained equilibrium optimal solutions are more superior in terms of diversification. That is, when the fuzziness of uncertain return rates is considered, the equilibrium optimal solution usually diversified the optimal solutions obtained by stochastic method.

As a consequence, when the exact probability distributions of return rates are unavailable, the proposed equilibrium optimization method provided an effective way to model practical portfolio selection problem.

Ankit Dangi (2012) addresses the problem that in practice, portfolio optimization faces challenges by virtue of varying mathematical formulations, parameters, business constraints and complex financial instruments. Empirical nature of data is no longer one-sided; thereby reflecting upside and downside trends with repeated yet unidentifiable cyclic behaviors potentially caused due to high frequency volatile movements in asset trades. Portfolio optimization under such circumstances is theoretically and computationally challenging. This work presents a novel mechanism to reach to an optimal solution by encoding a variety of optimal solutions in a solution bank to guide the search process with regard to the global investment objective formulation. It conceptualizes the role of individual solver agents that contribute optimal solutions to a bank of solutions, and a super-agent solver that learns from the solution bank, and, thus reflects a knowledge-based computationally guided agents approach to investigate, analyze and reach to optimal solution for informed investment decisions.

In order to avoid the slow convergence, this study relaxes constraints to soft constraints.

Markowitz Formulation (Soft Return Constraints)

$$\begin{aligned} \min \quad & \sigma_p^2 = x^T \Sigma x && \text{objective function} \\ \text{subject to} \quad & \sum_{i=1}^{n} r_i x_i \geq r_p && \text{return constraint} \\ & \sum_{i=1}^{n} x_i = 1 && \text{budget constraint} \\ & 0 \leq x_i \leq 1, \ \forall i && \text{long-only constraint} \end{aligned} \tag{2.2}$$

Where, the asset returns for the above formulation are the absolute return measures and absolute risk measures. The formulation considers arithmetic returns for the assets as the return measure and the standard deviation as the risk measure. Additionally, one could consider formulating the problem with other classes of parameters of estimating return measures that would address critical aspects of the financial markets which are significant to large financial institutional investors and public/private banks. These may include the Average Profit and Loss I.e. Average PnL (as an absolute value, or as a %age, or per each industrial sector, or as per a pre-specified time interval), and Compounded Annual Growth Rate (CAGR). Other critically relevant measures may include the relative return measures (incl. upward-downward movement capture ratio, or the upward-downward movement number/percentage ratio), or the absolute risk-adjusted return measures (incl. Sharpe ratio, Calmar ratio, Sterling ratio or Sortino ratio), or the relative risk-adjusted return measures.

Perrin & Roncalli (2019) realized the need for practical & established alternative solutions to the Markowitz model because even though it provided great computational ease, most of the models derived from it are not feasible from a financial perspective as they can only be utilized on a limited number of assets and require enormous computational power and resources, thereby raising the cost. Perrin & Roncalli aimed to find an alternative to the quadratic programming approach of the Markowitz model, they studied and analyzed four machine learning algorithms in order to determine their viability and extent of usefulness for large – scale portfolio optimization. The four algorithms analyzed were alternating direction method of multipliers, Dykstra's algorithm, coordinate descent and proximal gradient.

Perrin & Roncalli are focused more on the numerical implantation rather than just the quadratic objective function. They began with the selection of a non-quadratic objective function for asset allocation and then deriving the numerical solution of the selected non-quadratic function, in doing so they discovered that the coordinate descent algorithm is the fastest method for performing high-dimensional lasso regression on the other hand Dykstra's algorithm has been created to find the solution of restricted least squares regression. Since there is a strong correlation between Multi Variance Optimization (MVO) and linear regression, these algorithms could assist in solving MVO asset allocation models. Perrin & Roncalli later found out that even coordinate descent & Dykstra's algorithm has their limitation as they were unsuccessful in defining a standard framework, for which the need of alternating direction method of multipliers and proximal gradient methods was realized.

In short Perrin & Roncalli were successful in utilizing Multi Variance Optimization models with non-linear penalty functions such as the logarithmic barrier, etc. Also they devised an approach on handling the non-linear constraints such as volatility targeting, leverage limits, transaction costs, active share, etc. One of the most significant outcome of their study was combining the quadratic programming extensions and obataining an amalgamation of these four algorithms (CD, ADMM, PO and Dykstra) in order to utilize asset allocation models that cannot be cast into a Quadratic Programming form to achieve optimum results.

Ban, Karaoui & Lim (2013) sought to address the practical limitation of the portfolio optimization model when it is applied on real data by introducing a performance-based regularization (PBR) which acts as a limiter to sample variances of expected risk and return as it controls their stability which in turn helps in achieving a solution with lower estimation error and better performance. Ban, Karaoui & Lim introduce PBR with the main purpose to improve the out-of-sample performance of the result rather than focusing on the numerical stability (which was the approach of Perrin & Roncalli).

Ban, Karaoui & Lim mark four major milestones in their study. Firstly, the construction of a new portfolio optimization model by applying performance-based regularization (PBR) on mean-variance problems by introducing a new quartic polynomial constraint and analyze the convexity of the approximation and determine in quantifiable terms the effect of PBR on the result.

Secondly, they prove that performance-based regularization (PBR) can be utilized for robust optimization problems, the PBR constraint requires the expected return of the portfolio to be robust to all possible values of the mean vector falling within an ellipsoid, centered about the true mean, which not only solidifies the link between PBR and robust models but also justifies the empirical PBR standard in its own right.

Thirdly, Ban, Karaoui & Lim establish the fact that the solutions of Sample Average Approximation (SAA) and PBR are always asymptotically optimal because of the proven assumption that true solutions must be separated and identifiable, this is extremely significant due to the fact that most data-driven decisions cease to be asymptotically optimal when the number of observations is high. Lastly, they make a comparative study on SAA with respect to PBR and develop a new, performance-based extension of the k-fold cross-validation algorithm which is validated by computing the Sharpe ratio (prime practical performance metric for investment) rather than using the mean squared error.

In conclusion Ban, Karaoui & Lim are able to establish the fact that that PBR with performance-based cross-validation is highly adept at increasing the finite-sample performance of the data-driven portfolio decision compared to SAA proving without doubt that PBR is a superior modelling parameter.

Filos & Mandic (2018) have explored the concept of Reinforcement Learning, a branch of Machine Learning that efficiently solves sequential decision-making problem sets through direct interaction to the environment in a continuous manner. Filos & Mandic have used the aforementioned technique in developing portfolio optimization and asset allocation strategies.

The aim of this report is to determine the effectiveness of Reinforcement Learning models on asset allocation. After the selection of financial instruments, the model constructs a framework representation of the market & determine how to optimally allocate funds of a limited budget to those assets. The model is trained on both synthetic and real market data. Then the model performance is compared with standard portfolio management algorithms on a test data-set.

Furthermore, Filos & Mandic provide a thorough analysis of the modelling efficiency, performance advantages and expressive power of the trading models/agents developed (i.e., Deep Soft Recurrent Q-Network (DSRQN) and Mixture of Score Machines (MSM)), which have their roots both in traditional system identification as well as on context-independent agents. Filos & Mandic are also able to establish the fact that model-free reinforcement learning techniques not only reduce the memory size and computational complexity but are also able to standardize strategies across different financial instruments and markets, irrespective of the fact that they belong to different trading universe on which they have been trained.

The analysis and simulations conducted by Filos & Mandic confirm the superiority of universal model-free reinforcement learning agents over current portfolio management model in asset allocation strategies. The model obtained was found to outperformed all trading agents, in the S&P 500 and the EURO STOXX 50 markets. Lastly, model pre-training, data augmentation and simulations enabled robust training of deep neural network architectures, even with a limited number of available real market data.

CHAPTER 3 – RESEARCH METHODOLGY

Machine Learning for Portfolio Optimization is an Exploratory study to study various methods of Portfolio Optimization to arrive at the best method. Exploratory research is well-defined as a study used to inspect a problem which is not clearly defined. It is conducted to have an improved understanding of the prevailing issue, but will not provide conclusive results. For such a research, a researcher starts with a broad notion and uses this research as a mode to identify issues, that can be the focus for forthcoming research in the future. A central facet here is that the researcher must be enthusiastic to change his/her direction subject to the revelation of new data or insight. Such a research is usually carried out when the problem is at a preliminary stage. It is called as grounded theory approach or interpretive research as it used to explore answers to questions such as what, why and how.

3.1 EXPOLATORY STUDY METHODOLOGY

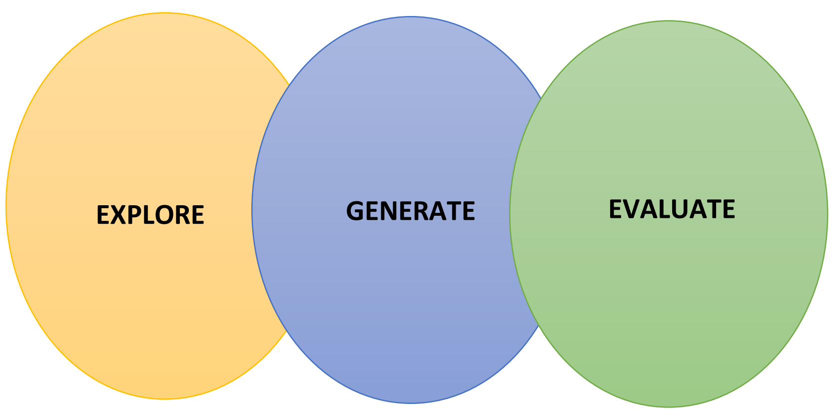

Fig 3.1: Exploratory study stages

STAGE I: EXPLORE

In this study initially the goals and scope of the research was explored. This phase aimed at exploring the main aspects of an under-researched problem, in this study it is portfolio optimization using Machine Learning techniques and Deep learning methods.

The historic and the ongoing methods of Portfolio optimization was studied. In-depth literature study led us to understanding the various methodologies used and also lead us to the understanding that though technical analysis is a very commonly used tool in portfolio optimization the extent of its utility is not fully capitalized. Machine Learning, Deep learning and other data intensive techniques find large scale utilization in trading and financial analytics but its utility in portfolio optimization is still limited.

Hence, the aim of this study was set as exploring the Machine Learning and Deep Learning techniques in Portfolio optimization.

STAGE II: GENERATE

This phase included generating a model for the study to be conducted. This stage included the identification of data points to be considered, the algorithm to be considered and the data analysis toolkit to be utilized.

The model chosen as fit for this exploratory study was cascading incremental algorithm testing using Python on to National Stock Exchange data of to 15 stocks over a period of 360 days. Average price of these stocks were chosen (to ensure reduced fluctuations in case of selection of opening prices or closing prices).

The techniques to be utilized was two pronged:

- Technical Analysis:

Technical analysis provides an outline for notifying investment management decisions by applying a supply and demand methodology to market prices. Fundamental principles underling the study of technical analysis are derived from the supposition that changes in the supply and demand of transacted securities affect their current market prices.

Tools of technical analysis are constructed on a background that seeks to achieve insight from the changes in forces of supply and demand. This structure has evolved over time from a totally graphic analysis to more quantitative techniques. Comparable to other analytical tools, technical analysis deploys a controlled, systematic approach that pursues to minimize the effect of interactive biases and emotion from the practice of speculation selection; subsequently, many institutional investment analysts, financial strategists, and portfolio administrators fuse technical research with other analytical methods, like quantitative, fundamental, and macroeconomic methods.

Independent or retail researchers have established the value of technical analysis, beginning with the validation of the momentum anomaly. Momentum, or relative strength in the lingo of technical analysts, has been functional from 1930s. It is now extensively accepted that relative strength analysis can benefit investment managers attain statistically and economically significant surplus or profits. Supplementary research has established the worth of other technical tools, as well as pattern investigation, moving averages, and other pointers.

Additional fresh research has spoken for the significance of technical analysis in the larger context of financial markets and commences to trace the connections among behavioral economics, individual players in financial markets, and the importance of technical analysis in studying the behavior of individual actors.

- Comparative Study

The comparative method of study is used in the initial stages of a branch of research. It can help the researcher to climb from the early stages of exploratory case studies to a more advanced level. The design of comparative study is pretty easy.

Objects are samples or cases which are similar in some respects (else, it would not be logical to compare them) but they diverge in some respects. These alterations become the center of the scrutiny. The goal is to discover why the cases are dissimilar: to reveal the common underlying arrangement which produces or allows such a variation.

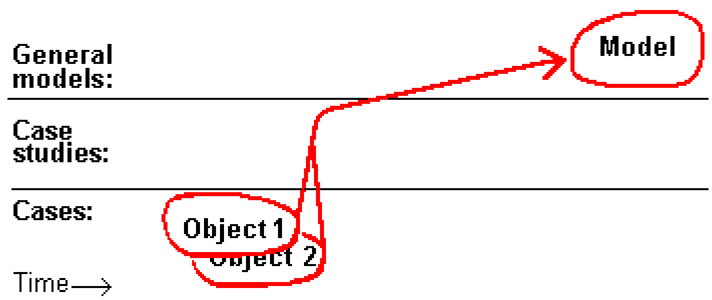

Fig 3.2: Comparative study Methodology

Contrast is one of the greatest efficient approaches for explaining or applying implicit knowledge or attitudes. This can be done, for example, by showing in parallel two slides of two slightly different objects or situations and by asking people to explain verbally their differences.

Fig 3.3: Research Lifecycle

3.1.1 Characteristics of Exploratory research

- They are unstructured researches
- It is low cost, collaborating and wide scoped.
- It will allow a researcher to respond to questions such what the problem is? What the goal of the study is? And what the topical scope of the study is?
- Commonly, there is no preceding research done or the current ones do not retort the problem accurately enough.
- It is a time intense and it needs perseverance and has perils related to it.
- It requires for the researcher to do a thorough research and study all the information.
- There is no set of guidelines for this research, as they are flexible, wide-ranging and dispersed.
- The research needs to have significance or worth. If the problem is not significant in the industry the study supported is vain.
- The study should have a few models which can support its results.
- Such a study generally produces qualitative data, but in definite cases quantitative data can be generalized for a larger sample.

3.1.2 Advantages of Exploratory research

- The investigator has a lot of tractability and can adapt to deviations as the study evolves.
- It is generally inexpensive.
- It supports laying the foundation of a study, which can lead to supplementary research.
- It allows the academic to understand at an initial stage, if the subject is worth capitalizing the time and resources.
- It can support other scholars to find out possible reasons for the issue, which can be supplementary studied in detail to figure out, which of them is the utmost cause of this problem.

3.2 Procedure

This study was fully devoted to the problem of portfolio optimization — it reviewed traditional mathematical methods for boosting portfolio, unsupervised, supervised machine learning approaches, reinforcement learning means and some more less traditional options.

3.2.1 The Optimization Machine

Regardless of the fact if investors are conscious of it or not, the choices investor make about portfolio structure express, their views (biases) about the characteristics of the asset (universal set) under deliberation. It is true whether the investor chooses to allot to reserves based on naive approaches, such as capitalization weights or equivalent weights; applying empirical methods like inverse volatility or variance; or deploying full-scale portfolio optimization. The question of whether the investor wishes to express the views consciously or unconsciously.

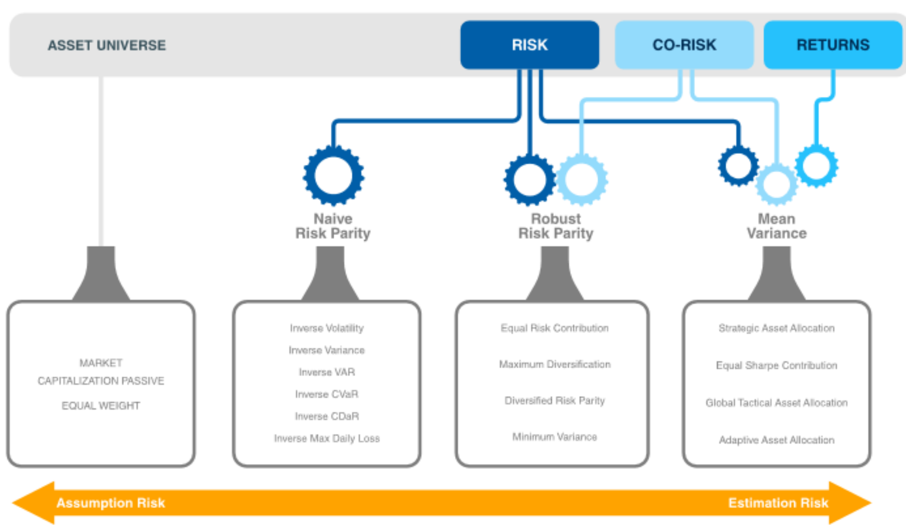

Fig 3.4: Optimization Machine

Fundamental Assumptions:

- Investors have inclinations that are well explained by mean-variance function. Investors desire to own the portfolio that capitalize on their expected return subject to a maximum tolerable portfolio volatility.

- Many investors barely care about volatility as compared to risks like “permanent loss of capital”, “maximum drawdown”, or “expected shortfall”. However, these alternative definitions of risk are well captured by mean and variance.
- A methodical approach is often rebalanced, a huge probable mean relative to volatility strongly implies a smaller risk of permanent loss, a smaller expected maximum drawdown, and a smaller expected shortfall

3.2.2 Classical optimization

Optimization problem requires maximizing or minimizing some function with respect to its parameters. In the case of this study, we maximize returns while minimizing the risk with respect to the amount of money we allocate on each asset in our portfolio.

Markowitz efficient frontier

Modern Portfolio Theory is based Markowitz's theory regarding maximizing the return investors can get as a result of the investment portfolio bearing in mind the risk associated with the said investments. Modern Portfolio Theory expects the investor to take into account how much the risk of a single investment can impact their complete portfolio., that every investor on the market is rational (is guided by the risk and return based on the market forces alone), circumvent risks and aim to maximize their anticipated returns.

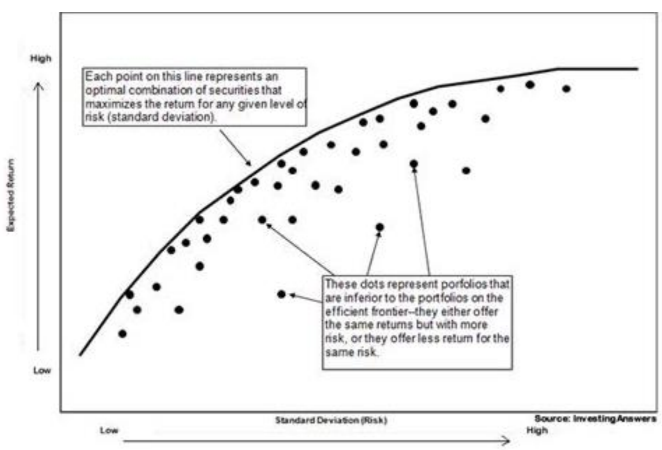

Fig 3.5: Markowitz Efficiency Frontier

The optimum decision could be established with capitalize on expected returns (deliberated from the preceding movements of the stock market assets) and diminishing accompanying risk (as unpredictability of the market assets). The proportion of anticipated return upon risk is known as Sharpe ratio and the portfolio with the highest Sharpe ratio can be established with a rather typical optimization toolkit.

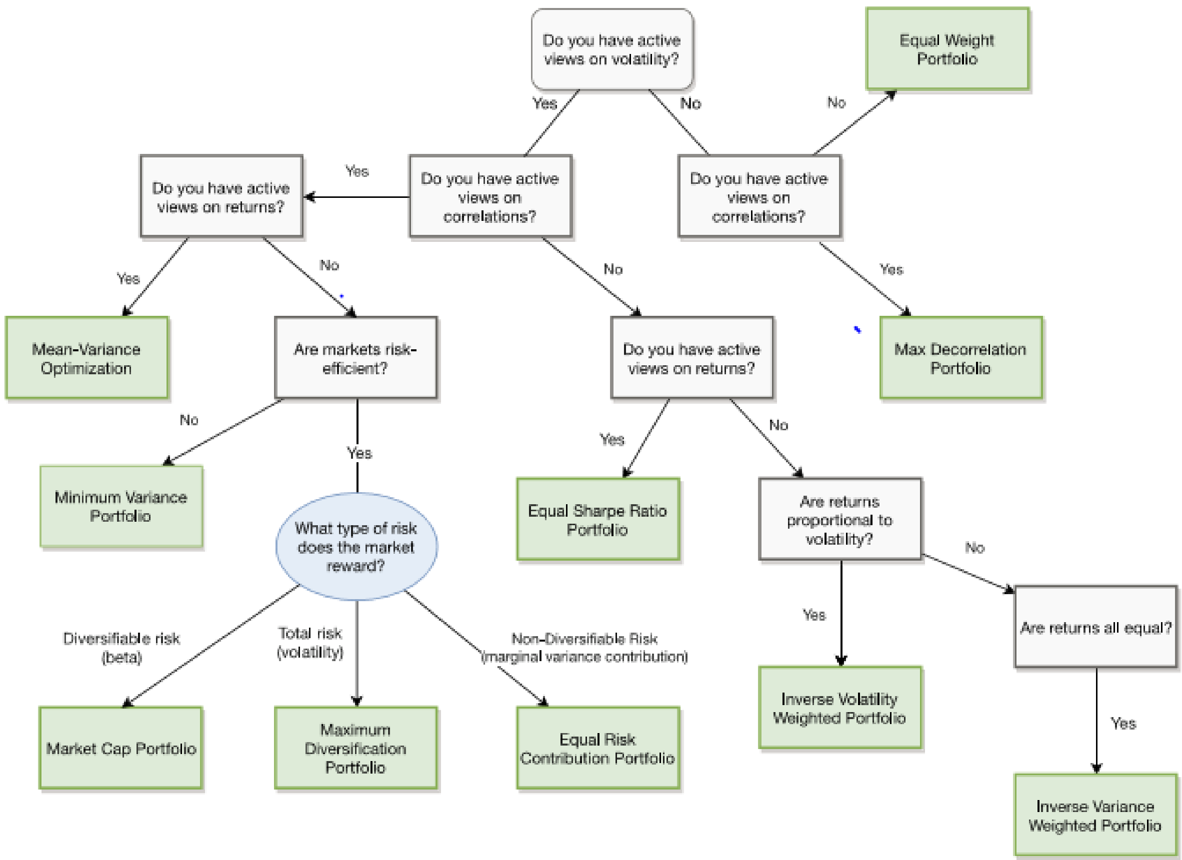

Fig 3.6: Portfolio Optimisation Decision Tree

There are a lot of different optimization criteria we might aim for. What if we even don't care about the expected returns and just want to minimize the risk?

Risk Based Optimization

The examination of risk-based optimization methods would be linked directly to a discussion of which risks are reimbursed in stock markets, provided risk is compensated in the first place.

Minimum Variance

All the investments have the identical anticipated return free of risk, stockholders in quest of maximum returns for minimum risk should concentrate exclusively on minimalizing risk. This is the unambiguous objective of the Minimum Variance portfolio.

$$w^{MV} = \arg\min\ w^T \cdot \Sigma \cdot w \tag{3.1}$$

(Haugen and Baker 1991) suggested giving up the relationship between risk and return, at least for equities. Stock market returns are not well expounded by beta. Rather, an adverse relationship among returns and volatility. In case of spurious association between risk and return, (Haugen and Baker 1991) suggested that a frequently altered long-only Minimum Variance portfolio will govern the capitalization weighted portfolio for bonds.

Maximum diversification

(Choueifaty and Coignard 2008) anticipated that marketplaces are risk-efficient, such that reserves will yield returns in ratio of total risk, as measured by volatility. This differs from CAPM, which adopts returns are comparative to non-diversifiable (i.e. systematic) risk. Unfailing with the opinion that returns are openly proportional to volatility, the Maximum Diversification mode of optimization proxies asset volatilities for yields in a maximum Sharpe ratio optimization.

$$w^{MD} = \arg\max \frac{w \times \sigma}{\sqrt{w^T \cdot \Sigma \cdot w}} \tag{3.2}$$

Maximum Decorrelation

It is defined by (Christoffersen et al. 2010) is strictly linked to Minimum Variance and Maximum Diversification, but relates to the occasion where a stockholder has faith in all assets have identical returns as well as volatility, but these are heterogeneous correlations.

$$w^{MDec} = \arg\min\, w^T \cdot A \cdot w \qquad (3.3)$$

Risk Parity

Both Minimum Variance as well as Maximum Diversification portfolios have mean-variance proficient under innate assumptions. Minimum Variance is competent if stocks have identical returns despite the fact that Maximum Diversification is effective if assets have related Sharpe ratios. However, both devices have the drawback that they are fairly focused on a small set of assets.

Inverse volatility and variance

Identical anticipated Sharpe ratios and a stockholder could not consistently approximate correlations, the optimum portfolio will be biased in proportion to the reverse of the assets' instabilities. When investments have identical expected yields and unidentified correlations, the Inverse Variance portfolio is mean-variance optimum.

3.2.3 Unsupervised learning

Portfolio optimization questions look a lot like unsupervised learning questions or representation learning tasks: having a set of stocks/bonds we need to group them into some "clusters" based on their cost-effectiveness and after allot more funds on the most predictive ones and not as much on the conflicting side.

Principal component analysis (PCA)

While there are a number of variables or features should be less than 10, then we are advised to perform PCA. PCA is an arithmetical technique which decreases the dimensions of the data and help us comprehend, visualise the data with lesser dimension compared to original data. As the name recounts PCA aids us calculate the Principal components of data. Principal components are fundamentally vectors that are linearly uncorrelated and sport a variance.

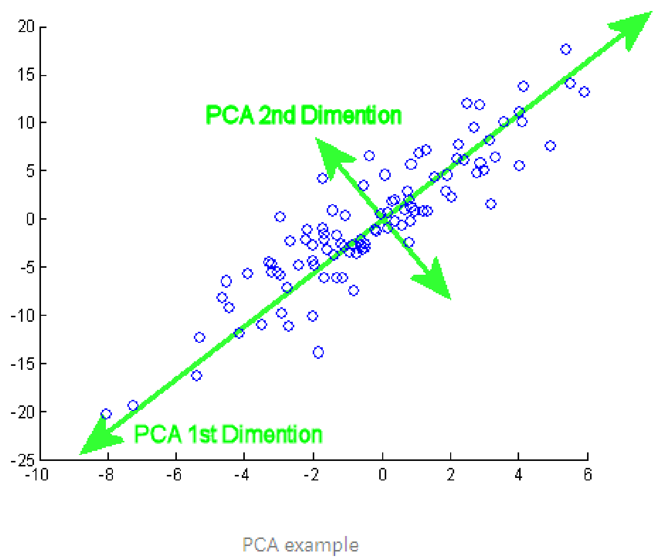

Fig 3.6 PCA decomposition

While dealing with data of financial markets the premier principal component serves as the closest estimate of the market, hence, selecting second or other components will give uncorrelated results to the stock market strategies, which is exactly what most of the investors desire.

Autoencoder risk

Autoencoder facilitates non-linear dimensionality reduction, is based on neural networks. They can "squeeze" data that's given as input in to a low-dimensional vector and after reinstating a input from this demonstration. Autoencoder could be capitalized in several ways for portfolio assortment, one of them is associated to the valuation of the risk supported by the specific asset: if some asset movement can't be reinstated well from the low-dimensional demonstration it's associated with greater risk.

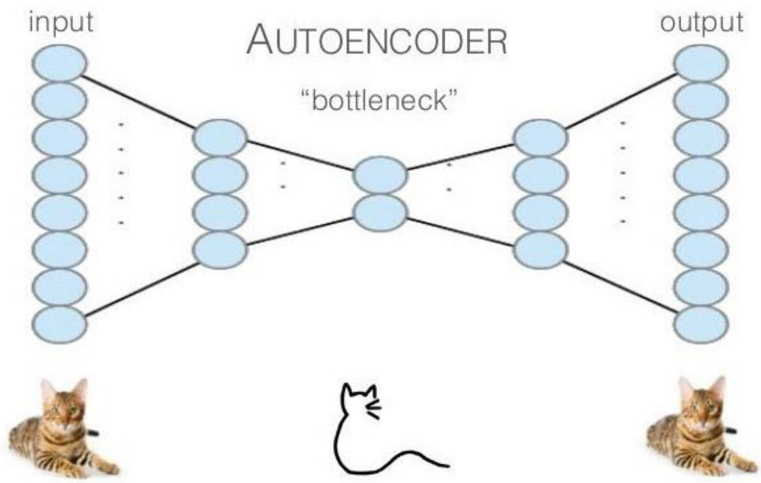

Fig 3.7: Autoencoder

Hierarchical risk parity

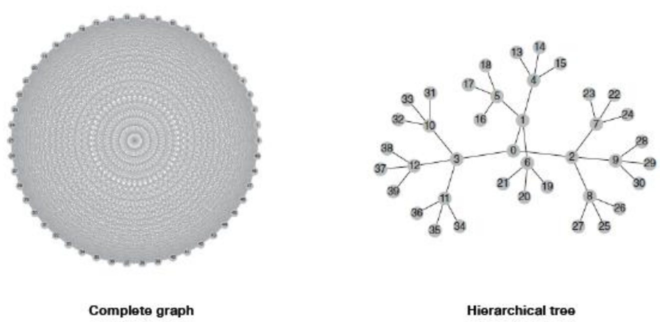

Fig 3.8: HRP

A covariance matrix of the stocks in the portfolio is a comprehensive graph. The optimization question, where we allot the risk rather than the capital funds. While occupied with portfolios of very massive scope — if we characterize links among assets geometrically, there would be in the form of the comprehensive plots, it is an **over-complication**. The solution lies again in unsupervised learning, but with the use of the hierarchical clustering algorithms applied to the covariance matrix. After finding clusters of the stocks, we can re-allocate risk over them recursively.

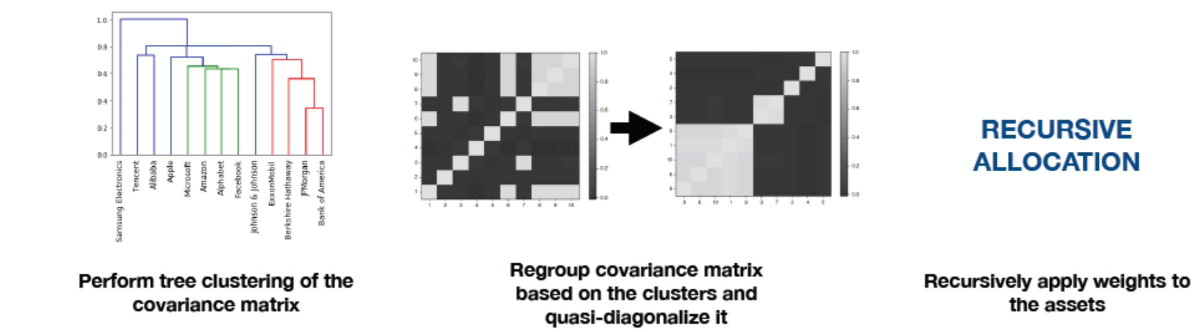

Fig 3.9: HRP Steps

3.2.4 Supervised learning

The optimization and unsupervised approaches appear good, but they have one main shortcoming: they just **exploit evidence about the historical** performance and associated asset movements and co-correlations without any norms for their future behaviour. There is no guarantee the future assets will move in a pattern same as in the past.

Forecasting weights

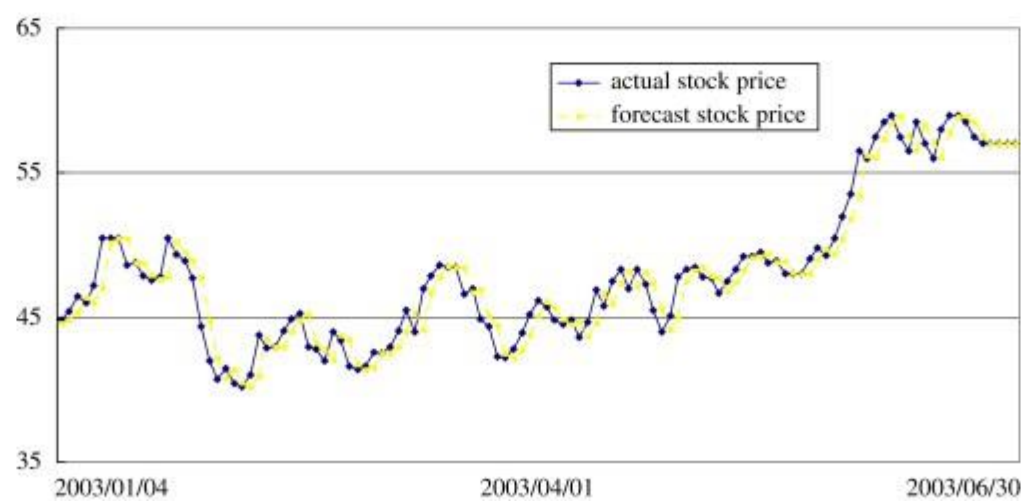

Fig 3.10: Forecasting Weights

The concept is pretty forthright: if we could use any model to predict the price movement in the future, we can use this forecast for allotting weight. Of course, we need to regularize these predictions so their summation is equal to unity, but this is rather a technical step and can be done with the single L1-normalization. We can deploy use the simplest predicting algorithm: exponential smoothing.

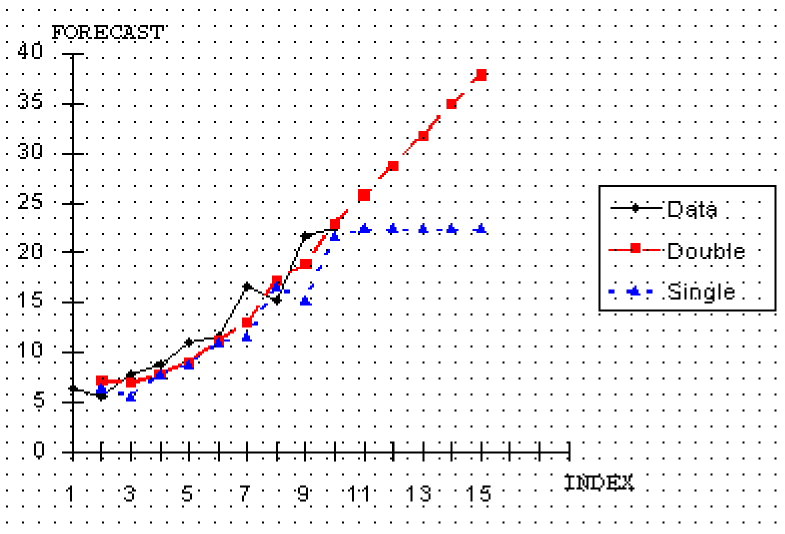

Fig 3.11: Smoothening

3.3 Python Architecture

Fig 3.12: Python Logo

The above mentioned business problem was solved using PYTHON. **Python** is an integrated, high-utility, generic language. Created by Guido van Rossum and released first 1991, Python's design focuses code readability making it easy to comprehend with its notable use in significant whitespace. Its language constructs and object centric approach aim to help programmers write pure, rational code for small and large-scale projects.

Python is dynamic and it supports multiple programing languages, including structured, object-oriented, and functional programing. Python is often described as a "batteries included" because of its inclusive standard library.

PyCharm is an integrated development environment (IDE) employed in programming, specially for the Python language. It is advanced by the Czech company JetBrains. It offers code examination, a graphical debugger, integrated unit tester, has full integration with version control systems (VCSes), and enables web development with Django as well as Data Science tools such as Anaconda.

Fig 3.12: Pycharm Logo

3.3.1 Libraries Used

Fig 3.13: Python Data analytics libraries

1. SciPy Optimize

It offers utilities for minimizing (or maximizing) objective functions, possibly subject to constrictions. It contains solvers for nonlinear complications (with provision for both global and local optimization procedures), linear programing, root finding, constrained and nonlinear least-squares and curve fitting.

- Minimize_scalar: minimizes a scalar function of one variable
- Minimize(fun, x0[, args, method, jac, hess, …]): minimizes a scalar function with one or more variables
- NonlinearConstraint(fun, lb, ub[, jac, …]): imposes constraint that are nonlinear on the variables.
- LinearConstraint(A, lb, ub[, keep_feasible]): imposes constraint that are linear on the variables.

2. scikit-learn

It is an easy and effective tool for extrapolative data analysis. Accessible to everybody, and recyclable in numerous circumstances. Constructed on NumPy, SciPy, and matplotlib. Open source and is commercially usable.

- It can be used for classification using algorithms such as random forest, support vector machines etc.

- It can help in forecasting a continuous series associated with the object. It uses algorithms such as nearest neighbors, random forest and SVR.
- *class* sklearn.decomposition.PCA(*n_components=None*, *copy=True*, *whiten=False*, *svd_solver='auto'*, *tol=0.0*, *iterated_power='auto'*, *random_state=None*): Even dimensionality reduction can be done to reduce the number of random variables. K-means, feature selection and PCA.

3. Pandas:

Python data analytics library is fast, powerful, flexible and facile to use open source data analysis and manipulation tool.

- **DataFrame** object for data manipulation with integrated indexing;
- **Reading and writing data** among data constructions and different formats: Microsoft Excel, SQL databases, CSV and text files and the fast HDF5 format;
- Smart **data alignment** and cohesive handling of **missing data**

4. Numphy:

NumPy is a very frequently used Python package, it stands for 'Numerical Python'. It is the essential library for scientific computing, which contains a prevailing n-dimensional array object, provide implements for integrating C, C++ etc.

Fig 3.14: Keras Library

5. Keras

Keras Python library that provides a clean and convenient way to create a range of deep learning models on top of Theano or TensorFlow. minimalist Python library for deep learning that can run on top of Theano or TensorFlow.

It was developed to make implementing deep learning models as fast and easy as possible for research and development.

It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. It is released under the permissive MIT license.

Keras was developed and maintained by François Chollet, a Google engineer using four guiding principles:

- **Modularity**: A model can be understood as a sequence or a graph alone. All the concerns of a deep learning model are discrete components that can be combined in arbitrary ways.
- **Minimalism**: The library provides just enough to achieve an outcome, no frills and maximizing readability.
- **Extensibility**: New components are intentionally easy to add and use within the framework, intended for researchers to trial and explore new ideas.
- **Python**: No separate model files with custom file formats. Everything is native Python.

6. Statsmodels

It is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. An extensive list of result statistics are available for each estimator. The results are tested against existing statistical packages to ensure that they are correct. The package is released under the open source Modified BSD (3-clause) license. The online documentation is hosted at statsmodels.org

7. Random

This module implements pseudo-random number generators for various distributions. For integers, there is uniform selection from a range. For sequences, there is uniform selection of a random element, a function to generate a random permutation of a list in-place, and a function for random sampling without replacement.

The functions supplied by this module are actually bound methods of a hidden instance of the random.random class. You can instantiate your own instances of Random to get generators that don't share state.

8. Matplotlib

It is a comprehensive library for creating static, animated, and interactive visualizations in Python.

- It helps Create develop publication quality plots with few lines of code and uses interactive figures such as zoom, pan and update.
- It allows customization of line styles, font properties, axes properties and can export and embed a number of file formats
- It extends tailored functionality to 3rd party languages

3.3.2 Program Architecture

- **Agent.py:** this file runs the tensor flow background to perform all the deep learning programs.

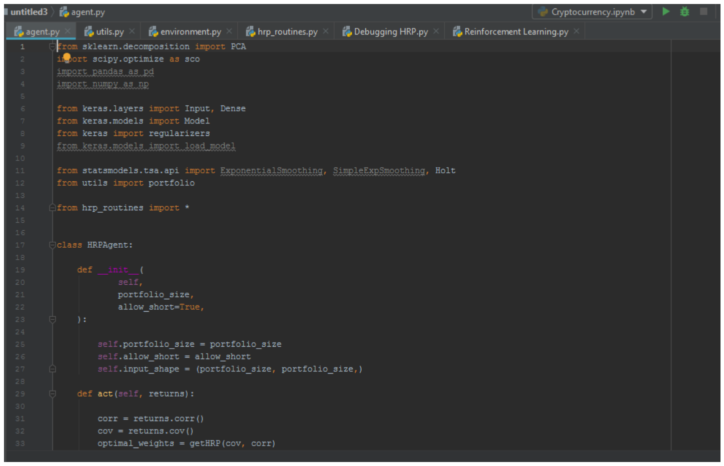

Fig 3.15: Agent.py

- **Utils.py:** this python script defines all the utilities that run to make all the algorithms work.

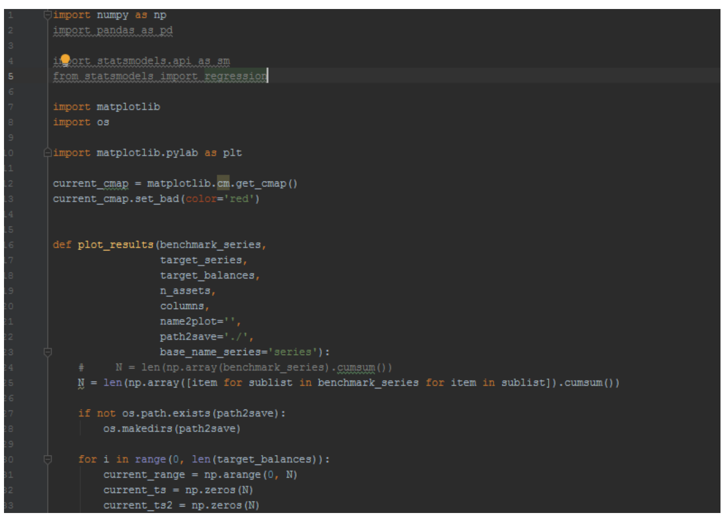

Fig 3.16: Utils.py

- **Environment.py**

This file calls the data files and builds a class variable that the code can work on. The class defines each of the labels and uses a pipeline to run the code.

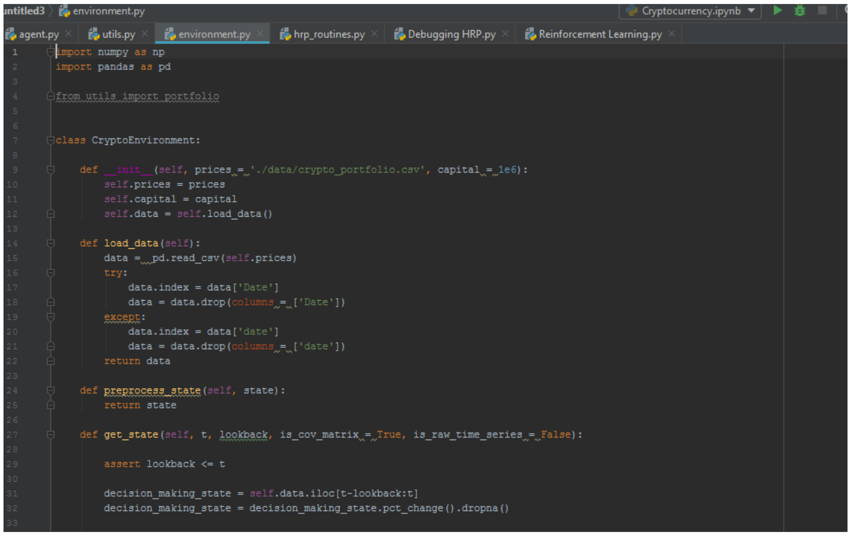

Fig 3.17: environment.py

- **HRP_routines.py**

Runs the HRP which is a unsupervised learning technique.

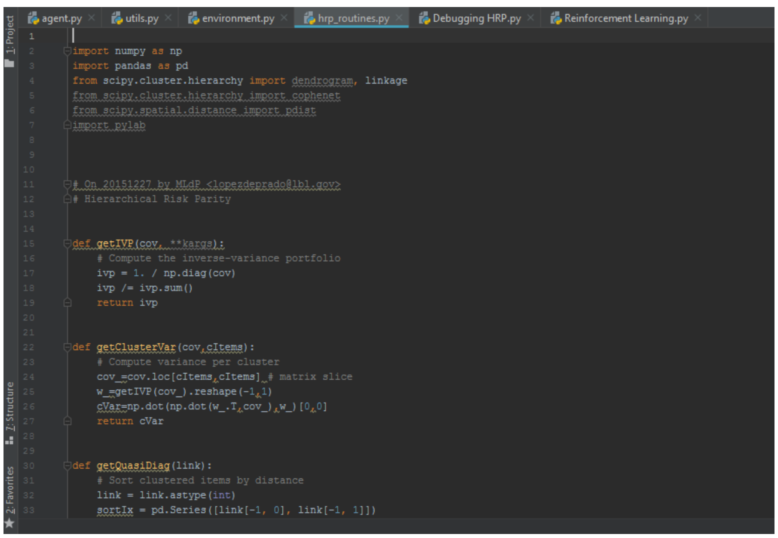

Fig 3.18: HRP-routines.py

CHAPTER 4 – RESULTS

4.1 Minimal Variance Portfolio (Markowitz)

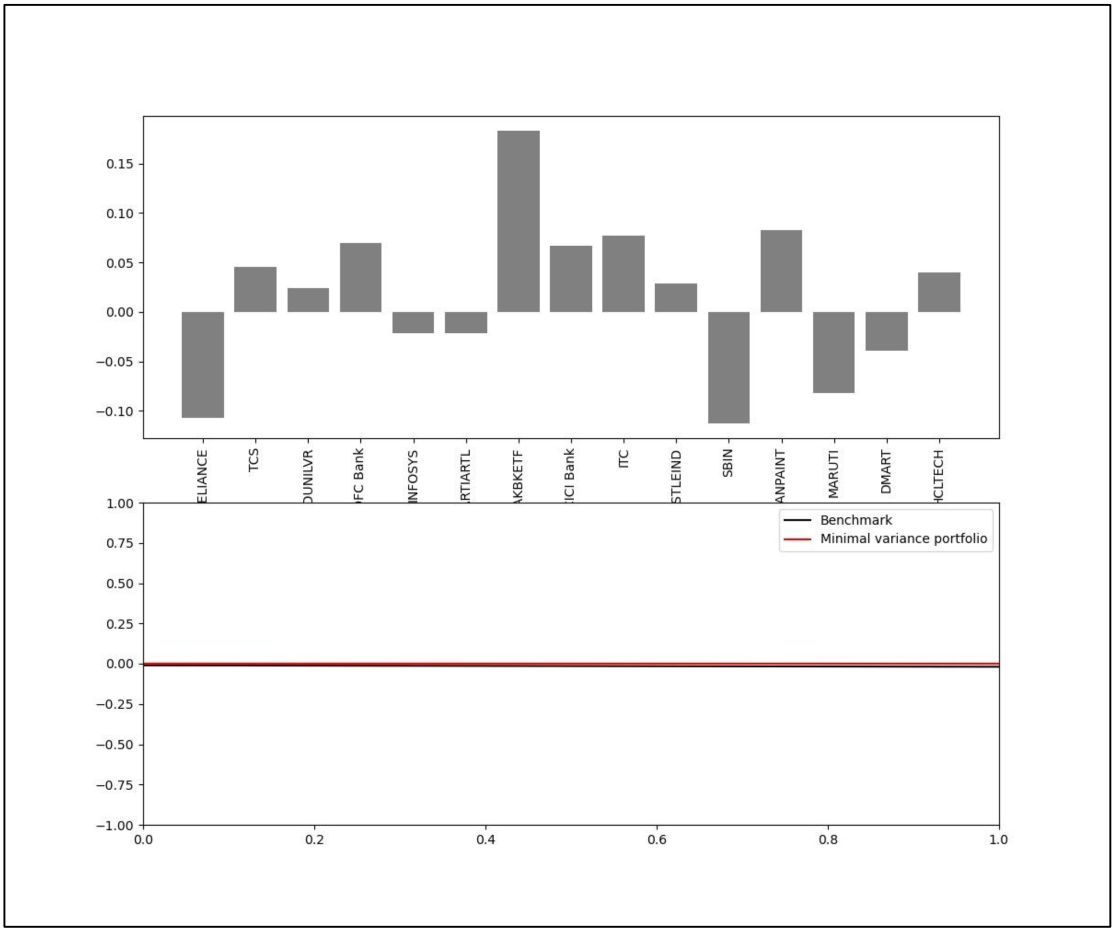

Fig 4.1 Minimal Variance Portfolio

Result obtained:

	MEAN RETURN	VOLATILITY	SHARPE RATIO	ALPHA	BETA
BENCHMARK	0.0005	0.00777	1.2545	0	1
MINIMAL VRIANCE	0.0004	0.0033	1.8305	0.0003	0.1297

Table 4.1: Minimal Variance Portfolio

4.2 Minimal Returns

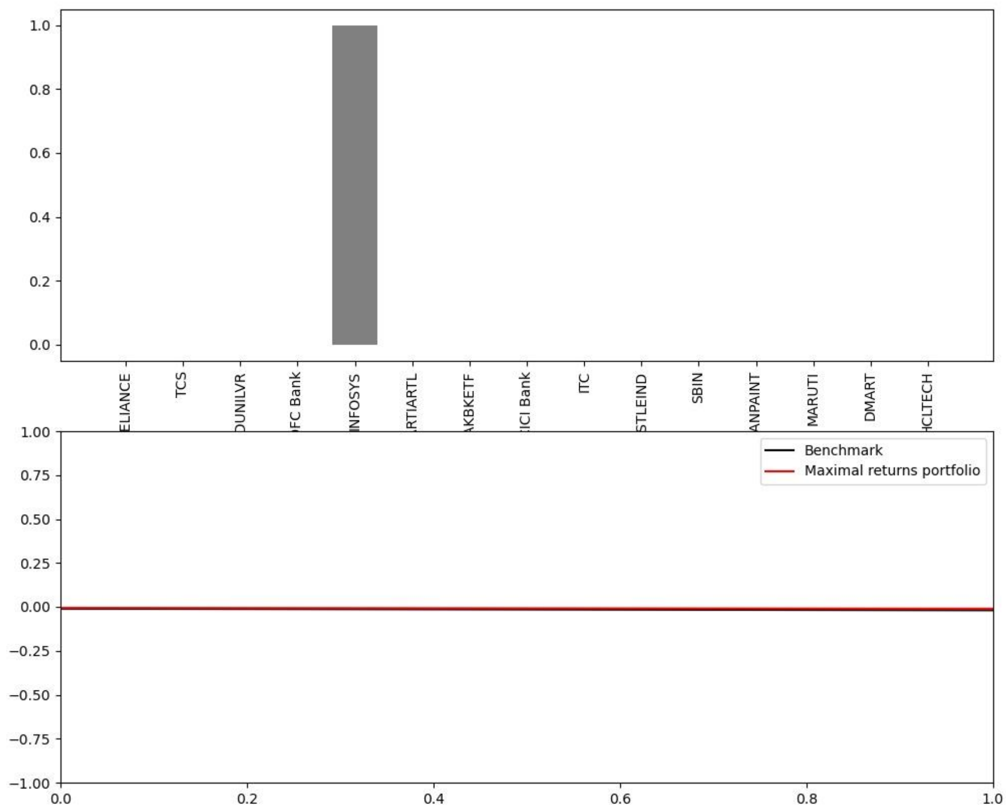

Fig 4.2: Minimal Return Portfolio

Results:

	MEAN RETURN	VOLATILITY	SHARPE RATIO	ALPHA	BETA
BENCHMARK	0.0005	0.00777	1.2545	0	1
MINIMAL RETURNS	0	0.0129	0.0033	0.0001	0.094

Table 4.2: Minimal Return Portfolio

4.3 Maximal Sharpe

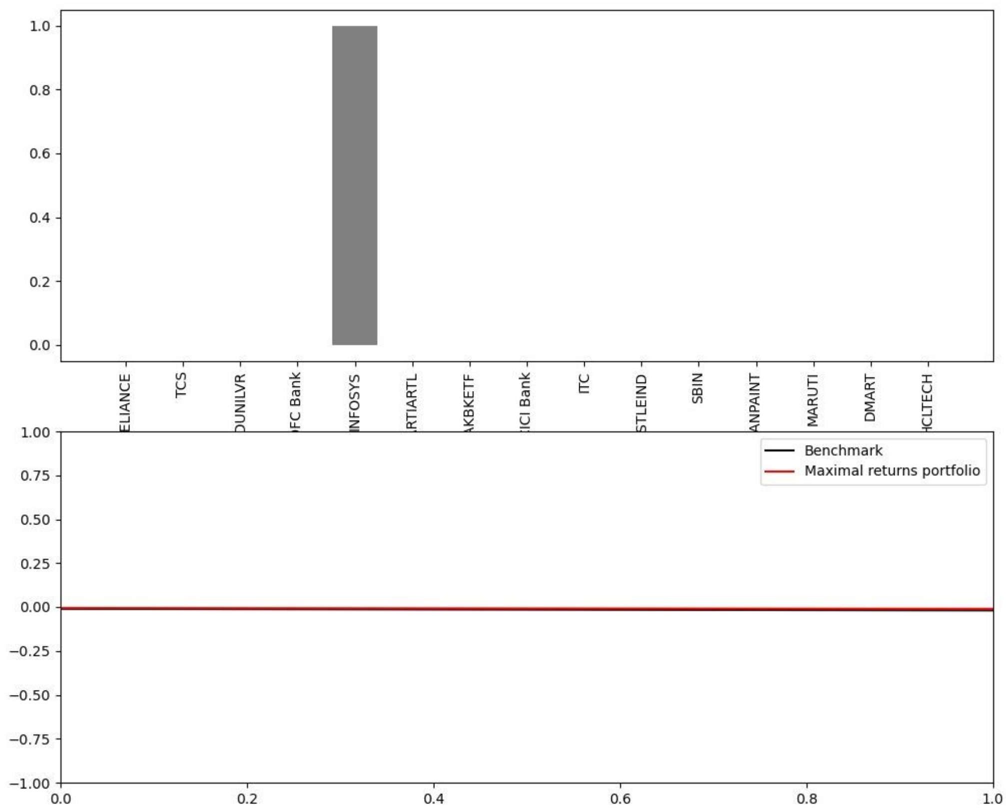

Fig 4.3: Maximal Sharpe Portfolio

Results:

	MEAN RETURN	VOLATILITY	SHARPE RATIO	ALPHA	BETA
BENCHMARK	0.0005	0.00777	1.2545	0	1
MAXIMAL SHARPE	0	0.004	0.1975	0	0.0275

Table 4.3: Maximal Sharpe Portfolio

4.4 Maximal Decorrelation

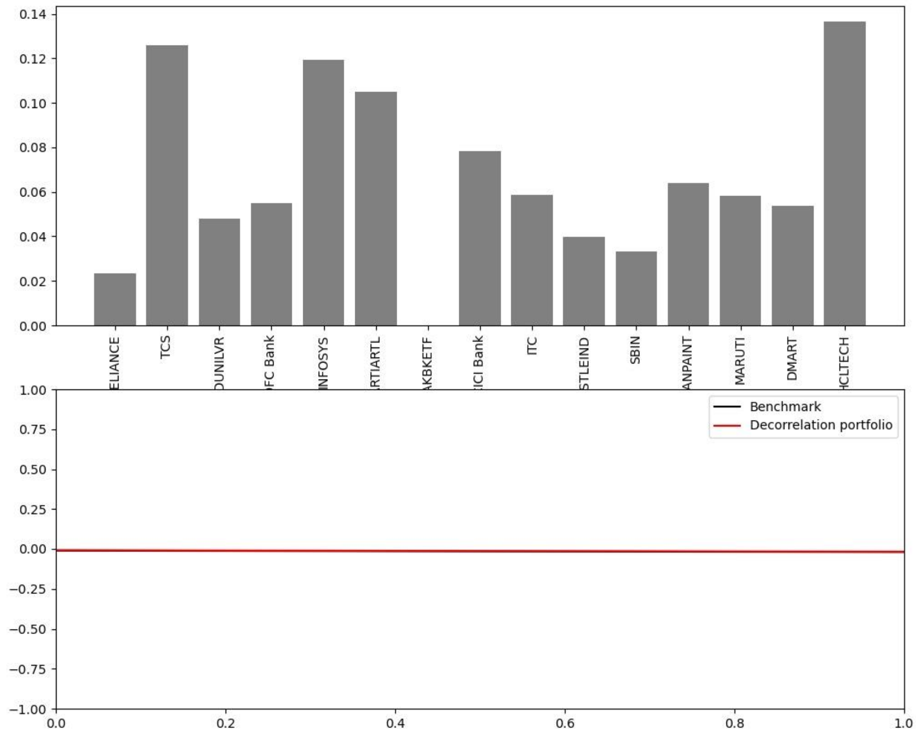

Fig 4.4: Maximal Decorrelation Portfolio

Results:

	MEAN RETURN	VOLATILITY	SHARPE RATIO	ALPHA	BETA
BENCHMARK	0.0005	0.00777	1.2545	0	1
MAXIMAL DECORRELATION	0.0005	0.0061	1.4572	0.0002	0.636

Table 4.4: Maximal Decorrelation Portfolio

The null hypothesis does not prove to be true for any variation of Markowitz.

4.5 Principal Component Analysis

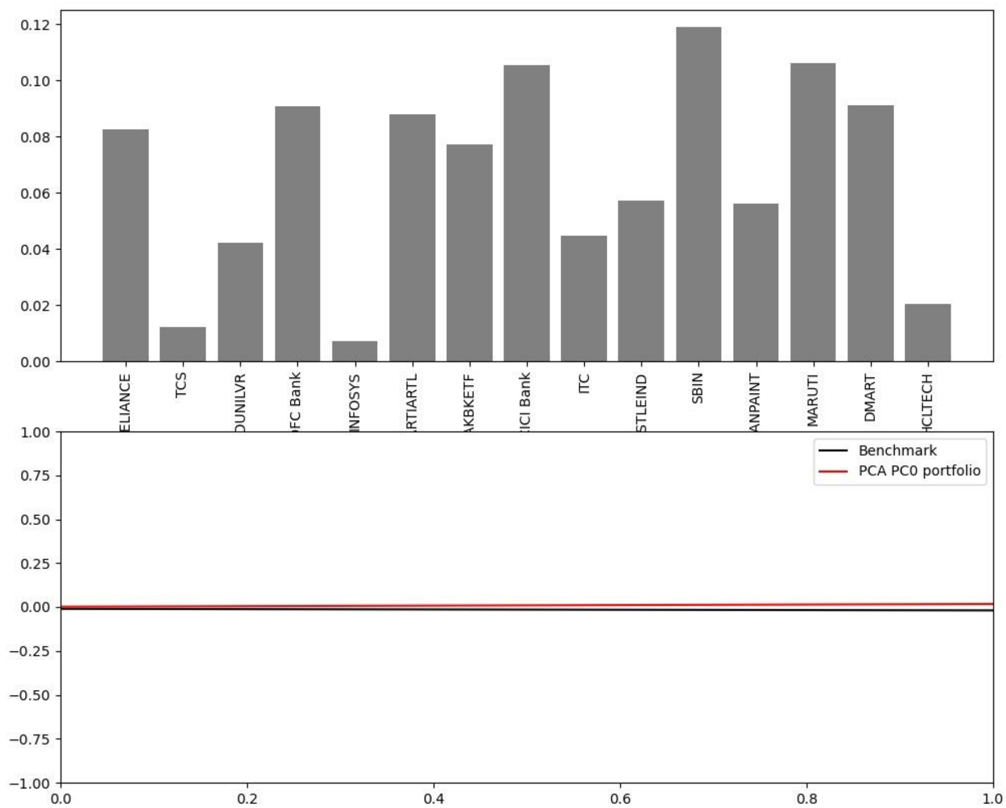

Fig 4.5: PCA Portfolio

Results:

The null hypothesis does not prove to be true.

	MEAN RETURN	VOLATILITY	SHARPE RATIO	ALPHA	BETA
BENCHMARK	0.0005	0.00777	1.2545	0	1
PCA	0.0006	0.0087	1.0608	0.0005	0.1431

Table 4.5: PCA Portfolio

4.6 Autoencoder

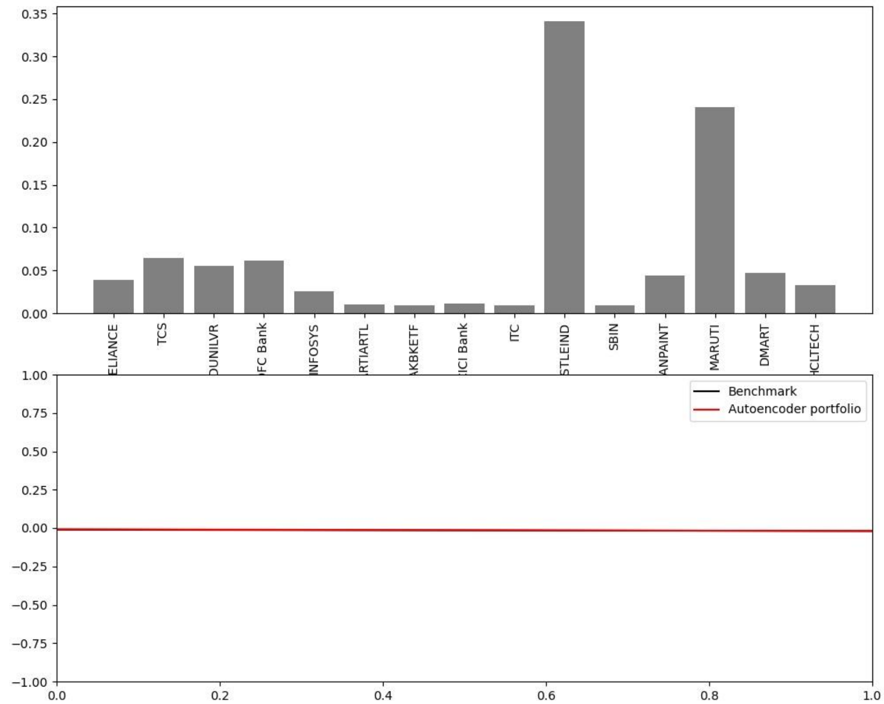

Fig 4.6: Autoencoder Portfolio

Results:

The null hypothesis does not prove to be true.

	MEAN RETURN	VOLATILITY	SHARPE RATIO	ALPHA	BETA
BENCHMARK	0.0005	0.00777	1.2545	0	1
AUTOENCODER	0.0002	0.009	0.5153	0.0003	0.0561

Table 4.6: Autoencoder Portfolio

4.7 Hierarchical Risk Parity Analysis

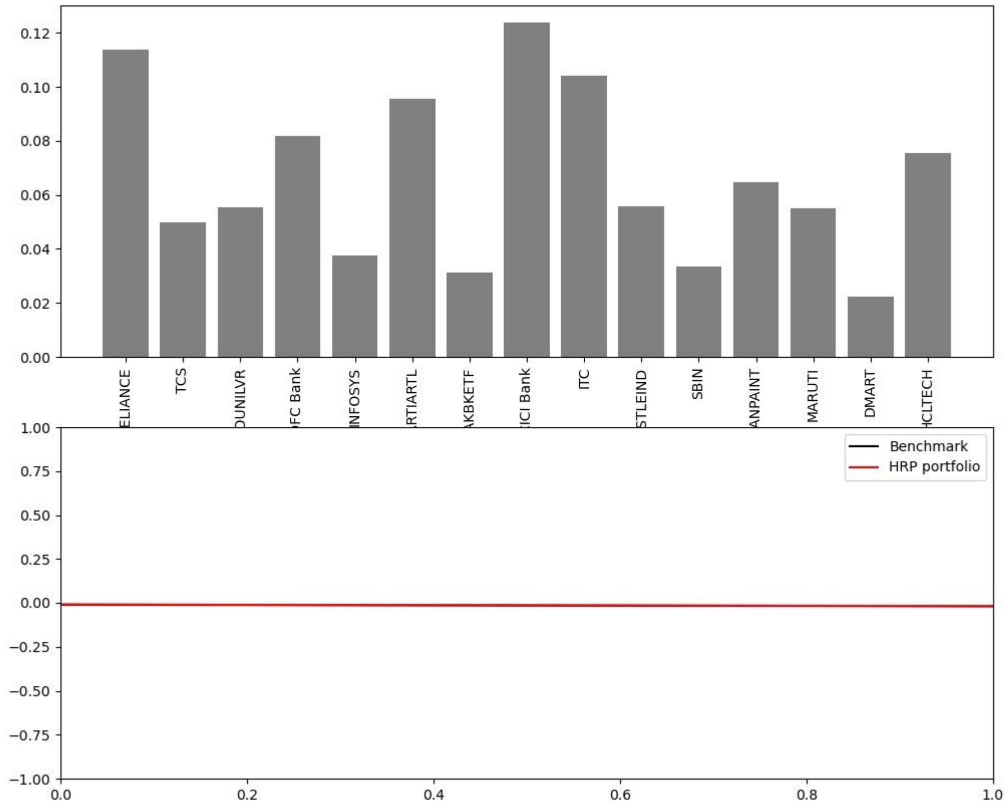

Fig 4.7: HRP Portfolio

Results:

The null hypothesis does not prove to be true.

	MEAN RETURN	VOLATILITY	SHARPE RATIO	ALPHA	BETA
BENCHMARK	0.0005	0.00777	1.2545	0	1
HRP	0.0004	0.0075	1.0421	0.0001	0.9571

Table 4.7: HRP Portfolio

4.8 Smoothening

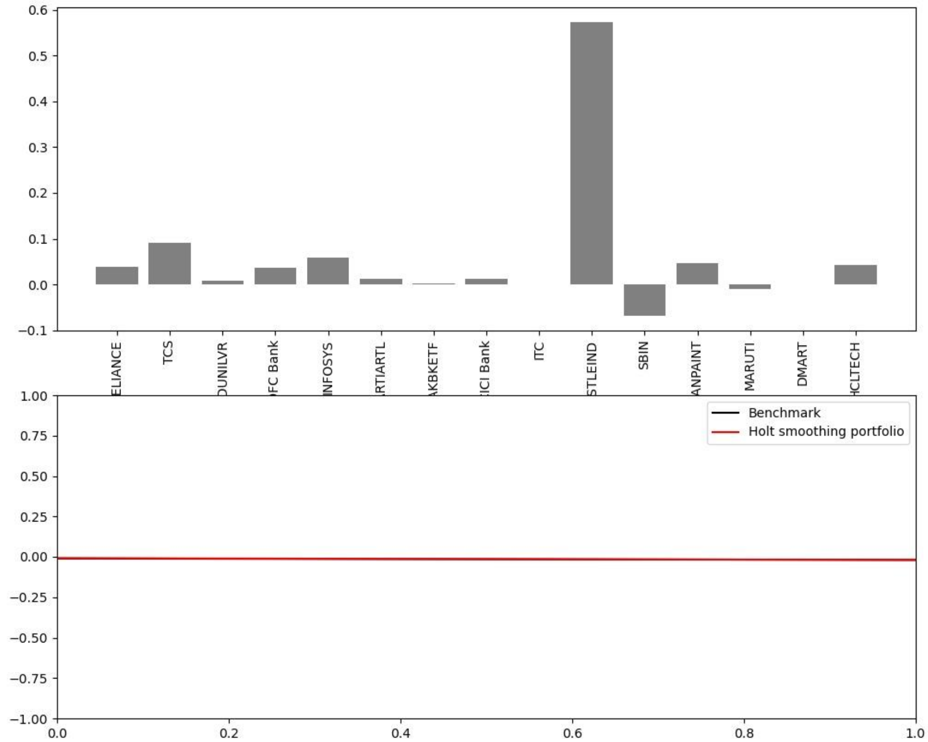

Fig 4.8: Smoothening Portfolio

Results:

The null hypothesis does not prove to be true.

	MEAN RETURN	VOLATILITY	SHARPE RATIO	ALPHA	BETA
BENCHMARK	0.0005	0.00777	1.2545	0	1
SMOOTHENING	0.0002	0.009	0.4921	0	0.3702

Table 4.8: Smoothening Portfolio

CHAPTER 5 – FINDING AND RECOMMENDATION

	BENCH MARK	MINIMAL VARIANCE	MINIMAL RETURNS	MAXIMAL SHARPE	MAXIMAL DECORREL ATION	PCA	HRP	AUTOEN CODER	SMOOTH ENING
MEAN RETURN	**0.0005**	0.0004	0	0	0.0005	0.0006	0.0004	0.0002	0.0002
VOLATIL ITY	**0.0078**	0.0033	0.0129	0.004	0.0061	0.0087	0.0075	0.009	0.009
SHARPE RATIO	**1.2545**	1.8305	0.0033	0.1975	1.4572	1.0608	1.0421	0.5153	0.4921
ALPHA	**0**	0.0003	0.0001	0	0.0002	0.0005	0.0001	0.0003	0
BETA	**1**	0.1297	0.094	0.0275	0.636	0.1431	0.9571	0.561	0.3702

Table 5.1: Portfolio Results

The Markowitz techniques

In case of Minimal Variance, Minimal Return, Maximal Sharpe:

- Cater to only a single aim
- Fail on all other parameters

It is fair to conclude that it is not the best solution available to us.

But surprisingly, **Maximal Decorrelation** fairs well on fronts:

- **Lower Alpha** shows that the gap between the actual and expected returns is low, showing an improved prediction model.
- **Lower Beta** of portfolio built based on maximal decorrelation denotes that the performance of the portfolio is not dependent on the market fluctuations i.e. sensitivity to market volatility is low.
- **Lower Volatility** is good as it makes the portfolio less risky

- **Higher Sharpe** indicates that fund in the given portfolio is risk-adjusted, which again is a positive indicator, which caters to all the risk and is risk-adjusted
- **Higher Returns**

Unsupervised Learning Techniques

In case of PCA fares well on all fronts:

- **Lower Alpha** shows that the gap between the actual and expected returns is low, showing an improved prediction model.
- **Lower Beta** of portfolio built based on PCA denotes that the performance of the portfolio is not dependent on the market fluctuations i.e. sensitivity to market volatility is low.
- **Lower Volatility** is good as it makes the portfolio less risky
- **Higher Sharpe** indicates that fund in the given portfolio is risk-adjusted, which again is a positive indicator, which caters to all the risk and is risk-adjusted
- **Higher Returns** more than the benchmark

Though Autoencoder and HRP are better methods and often seen as upgrades to PCA in our case they do not seem to make the cut.

Supervised Learning

The method of supervised learning used here, Smoothening does not fare well on the following counts:

- Lower Returns
- Lower Sharpe indicating that the portfolio is not adjusted to the risks.

A more advanced method like Reinforced Learning can overcome this limitation.

Conclusions:

- There is no one solution to your portfolio woes.

- Complex algorithms may not always mean better results.
- Definite goal setting is mandatory for Markowitz to work your way.
- The results of this model may or may not align for other market options like securities, ETF or cryptocurrency.

CHAPTER 6 – LIMITATION

DATA SET CONSTRAINTS

- The module and all the algorithms were tested on the same data set. The results could have been validated better if we could fit the given model on a different data set.
- The data set pans over 360 days, a data set over a longer time frame would have had different results.
- The data used is unidimensional is bound to only prices. Exploring the volume traded with the price could have led to different results.
- This study is limited to the top 15 stocks of the NSE. A study of other stocks could have resulted to variable results.

MODEL CONSTRAINTS

- The study is limited to the 6 methods mentioned above.
- Study of Reinforced learning technique would help explore newer patterns
- Deep Learning and Q learning techniques were not explored

MARKET CONSTRAINTS

- 2018-2019 have not been the greatest years in terms of the growth
- The slowdown has set in and this has effects on the data set chosen

REFERENCES

1. Khan,M.Y.(2018).*Indian Financial System* (10th ed.). Chennai : McGraw Hill Education Pvt. Ltd.
2. Security Lending Scheme (1997) .Retrieved from Securities & Exchange Board of India : https://www.sebi.gov.in/otherentry/jun-2002/securities-lending-scheme .
3. Grinold,Richard.C.,and Kahn ,Ronald.N.(1999). *Active Portfolio Management* (2nd ed.). New York : McGraw Hill Education Pvt. Ltd.
4. Jordan, R.J. and Fischer, D.E. (2007). *Security Analysis and Portfolio Management* (6th ed.). Mumbai : Prentice Hall of India Pvt Ltd.
5. Bodie, Z., Kane, A., Marcus, A.J. &Mohanty, P. (2015). *Investments* (11th ed.). Chennai: Tata McgGraw Hill Pvt Ltd.
6. Ivanova,M., Dospatliev,L.(2017). Application of Markowitz Portfolio Optimization on Bulgarian Stock Market. International Journal of Pure and Applied Mathematics(2),291-307. Retrieved from http://www.ijpam.eu
7. Wang,Ye., Chen, Yanju., Liu, YanKui.(2016). Modeling Portfolio Optimization Problem by Probability-Credibility Equilibrium Risk Criterion. Retrieved from Hindawi Publishing Corporation http://dx.doi.org/10.1155/2016/9461021
8. Dangi,Ankit.(2012). Financial Portfolio Optimization : Computationally guided agents to investigate, analyse and invest. Retrieved from Research Gate https://www.researchgate.net/publication/234140428
9. Perrin,Sarah., Roncalli, Thierry.(2019,June). Machine Learning Optimization Algorithms & Portfolio Allocation. Retrieved from Research Gate https://www.researchgate.net/publication/334647034
10. Ban, Gah-Yi., Karoui, Noureddine El., Lim, Andrew E.B.(2013). Machine Learning and Portfolio Optimization. Management Science Articles(3) 239-318. Retrieved from Semantic Scholar https://www.semanticscholar.org/paper/Machine-Learning-and-Portfolio-Optimization-Ban-Karoui
11. Filos, Angelos., Mandic, Danilo.(2018,June 20). Reinforcement Learning for Portfolio Management. Retrieved from https://arxiv.org/abs/1909.09571

CHAPTER 8 – ANNEXURE

1. Utils.py

```
import numpy as np
import pandas as pd

import statsmodels.api as sm
from statsmodels import regression

import matplotlib
import os

import matplotlib.pylab as plt

current_cmap = matplotlib.cm.get_cmap()
current_cmap.set_bad(color='red')


def plot_results(benchmark_series,
                 target_series,
                 target_balances,
                 n_assets,
                 columns,
                 name2plot='',
                 path2save='./',
                 base_name_series='series'):
    #     N = len(np.array(benchmark_series).cumsum())
    N = len(np.array([item for sublist in benchmark_series for item in
sublist]).cumsum())

    if not os.path.exists(path2save):
        os.makedirs(path2save)

    for i in range(0, len(target_balances)):
        current_range = np.arange(0, N)
        current_ts = np.zeros(N)
        current_ts2 = np.zeros(N)

        ts_benchmark = np.array([item for sublist in benchmark_series[:i
+ 1] for item in sublist]).cumsum()
        ts_target = np.array([item for sublist in target_series[:i + 1]
for item in sublist]).cumsum()

        t = len(ts_benchmark)
        current_ts[:t] = ts_benchmark
        current_ts2[:t] = ts_target

        current_ts[current_ts == 0] = ts_benchmark[-1]
        current_ts2[current_ts2 == 0] = ts_target[-1]

        plt.figure(figsize=(12, 10))

        plt.subplot(2, 1, 1)
        plt.bar(np.arange(n_assets), target_balances[i], color='grey')
        plt.xticks(np.arange(n_assets), columns, rotation='vertical')

        plt.subplot(2, 1, 2)
        plt.colormaps = current_cmap
```

```
        plt.plot(current_range[:t], current_ts[:t], color='black',
label='Benchmark')
        plt.plot(current_range[:t], current_ts2[:t], color='red',
label=name2plot)
        plt.plot(current_range[t:], current_ts[t:], ls='--', lw=.1,
color='black')
        plt.autoscale(False)
        plt.ylim([-1, 1])
        plt.legend()
        plt.savefig(path2save + base_name_series + str(i) + '.jpg')


def portfolio(returns, weights):
    weights = np.array(weights)
    rets = returns.mean() * 252
    covs = returns.cov() * 252
    P_ret = np.sum(rets * weights)
    P_vol = np.sqrt(np.dot(weights.T, np.dot(covs, weights)))
    P_sharpe = P_ret / P_vol
    return np.array([P_ret, P_vol, P_sharpe])


def sharpe(R):
    r = np.diff(R)
    sr = r.mean() / r.std() * np.sqrt(252)
    return sr


import statsmodels.api as sm
from statsmodels import regression


def print_stats(result, benchmark):
    sharpe_ratio = sharpe(np.array(result).cumsum())
    returns = np.mean(np.array(result))
    volatility = np.std(np.array(result))

    X = benchmark
    y = result
    x = sm.add_constant(X)
    model = regression.linear_model.OLS(y, x).fit()
    alpha = model.params[0]
    beta = model.params[1]

    return np.round(np.array([returns, volatility, sharpe_ratio, alpha,
beta]), 4).tolist()
```

2. Agent.py

```
from sklearn.decomposition import PCA
import scipy.optimize as sco
import pandas as pd
import numpy as np

from keras.layers import Input, Dense
from keras.models import Model
from keras import regularizers
from keras.models import load_model

from statsmodels.tsa.api import ExponentialSmoothing,
SimpleExpSmoothing, Holt
from utils import portfolio

from hrp_routines import *


class HRPAgent:

    def __init__(
            self,
            portfolio_size,
            allow_short=True,
    ):

        self.portfolio_size = portfolio_size
        self.allow_short = allow_short
        self.input_shape = (portfolio_size, portfolio_size,)

    def act(self, returns):

        corr = returns.corr()
        cov = returns.cov()
        optimal_weights = getHRP(cov, corr)

        if self.allow_short:
            optimal_weights /= sum(np.abs(optimal_weights))
        else:
            optimal_weights += np.abs(np.min(optimal_weights))
            optimal_weights /= sum(optimal_weights)

        return optimal_weights


class AutoencoderAgent:

    def __init__(
            self,
            portfolio_size,
            allow_short=True,
            encoding_dim=25
    ):

        self.portfolio_size = portfolio_size
        self.allow_short = allow_short
        self.encoding_dim = encoding_dim

    def model(self):
        input_img = Input(shape=(self.portfolio_size,))
        encoded = Dense(self.encoding_dim, activation='relu',
```

```
kernel_regularizer=regularizers.l2(1e-6))(input_img)
        decoded = Dense(self.portfolio_size, activation='linear',
kernel_regularizer=regularizers.l2(1e-6))(encoded)
        autoencoder = Model(input_img, decoded)
        autoencoder.compile(optimizer='adam', loss='mse')
        return autoencoder

    def act(self, returns):
        data = returns
        autoencoder = self.model()
        autoencoder.fit(data, data, shuffle=False, epochs=25,
batch_size=32, verbose=False)
        reconstruct = autoencoder.predict(data)

        communal_information = []

        for i in range(0, len(returns.columns)):
            diff = np.linalg.norm((returns.iloc[:, i] - reconstruct[:,
i]))  # 2 norm difference
            communal_information.append(float(diff))

        optimal_weights = np.array(communal_information) /
sum(communal_information)

        if self.allow_short:
            optimal_weights /= sum(np.abs(optimal_weights))
        else:
            optimal_weights += np.abs(np.min(optimal_weights))
            optimal_weights /= sum(optimal_weights)

        return optimal_weights


class SmoothingAgent:

    def __init__(
            self,
            portfolio_size,
            allow_short=True,
            forecast_horizon=252,
    ):

        self.portfolio_size = portfolio_size
        self.allow_short = allow_short
        self.forecast_horizon = forecast_horizon

    def act(self, timeseries):

        optimal_weights = []

        for asset in timeseries.columns:
            ts = timeseries[asset]
            fit1 = Holt(ts).fit()
            forecast = fit1.forecast(self.forecast_horizon)
            prediction = forecast.values[-1] - forecast.values[0]
            optimal_weights.append(prediction)

        if self.allow_short:
            optimal_weights /= sum(np.abs(optimal_weights))
        else:
            optimal_weights += np.abs(np.min(optimal_weights))
            optimal_weights /= sum(optimal_weights)
```

```
        return optimal_weights


class PCAAgent:

    def __init__(
            self,
            portfolio_size,
            pc_id=0,
            pca_max=10,
            allow_short=False,
    ):

        self.portfolio_size = portfolio_size
        self.allow_short = allow_short
        self.input_shape = (portfolio_size, portfolio_size,)
        self.pc_id = pc_id
        self.pc_max = pca_max

    def act(self, returns):
        C = self.pc_max
        pca = PCA(C)
        returns_pca = pca.fit_transform(returns)
        pcs = pca.components_

        pc1 = pcs[self.pc_id, :]

        if self.allow_short:
            optimal_weights = pc1 / sum(np.abs(pc1))
        else:
            optimal_weights += np.abs(np.min(optimal_weights))
            optimal_weights /= sum(optimal_weights)

        return optimal_weights


class MaxReturnsAgent:

    def __init__(
            self,
            portfolio_size,
            allow_short=False,
    ):

        self.portfolio_size = portfolio_size
        self.allow_short = allow_short
        self.input_shape = (portfolio_size, portfolio_size,)

    def act(self, returns):

        def loss(weights):
            return -portfolio(returns, weights)[0]

        n_assets = len(returns.columns)

        if self.allow_short:
            bnds = tuple((-1.0, 1.0) for x in range(n_assets))
            cons = ({'type': 'eq', 'fun': lambda x: 1.0 -
np.sum(np.abs(x))})
        else:
            bnds = tuple((0.0, 1.0) for x in range(n_assets))
            cons = ({'type': 'eq', 'fun': lambda x: 1.0 - np.sum(x)})
```

```
        opt_S = sco.minimize(
            loss,
            n_assets * [1.0 / n_assets],
            method='SLSQP', bounds=bnds,
            constraints=cons)

        optimal_weights = opt_S['x']

        # sometimes optimization fails with constraints, need to be
fixed by hands
        if self.allow_short:
            optimal_weights /= sum(np.abs(optimal_weights))
        else:
            optimal_weights += np.abs(np.min(optimal_weights))
            optimal_weights /= sum(optimal_weights)

        return optimal_weights


class MinVarianceAgent:

    def __init__(
            self,
            portfolio_size,
            allow_short=False,
    ):

        self.portfolio_size = portfolio_size
        self.allow_short = allow_short
        self.input_shape = (portfolio_size, portfolio_size,)

    def act(self, returns):

        def loss(weights):
            return portfolio(returns, weights)[1] ** 2

        n_assets = len(returns.columns)

        if self.allow_short:
            bnds = tuple((-1.0, 1.0) for x in range(n_assets))
            cons = ({'type': 'eq', 'fun': lambda x: 1.0 -
np.sum(np.abs(x))})
        else:
            bnds = tuple((0.0, 1.0) for x in range(n_assets))
            cons = ({'type': 'eq', 'fun': lambda x: 1.0 - np.sum(x)})

        opt_S = sco.minimize(
            loss,
            n_assets * [1.0 / n_assets],
            method='SLSQP', bounds=bnds,
            constraints=cons)

        optimal_weights = opt_S['x']

        # sometimes optimization fails with constraints, need to be
fixed by hands
        if self.allow_short:
            optimal_weights /= sum(np.abs(optimal_weights))
        else:
            optimal_weights += np.abs(np.min(optimal_weights))
            optimal_weights /= sum(optimal_weights)

        return optimal_weights
```

```python
class MaxSharpeAgent:

    def __init__(
            self,
            portfolio_size,
            allow_short=False,
    ):

        self.portfolio_size = portfolio_size
        self.allow_short = allow_short
        self.input_shape = (portfolio_size, portfolio_size,)

    def act(self, returns):

        def loss(weights):
            return -portfolio(returns, weights)[2]

        n_assets = len(returns.columns)

        if self.allow_short:
            bnds = tuple((-1.0, 1.0) for x in range(n_assets))
            cons = ({'type': 'eq', 'fun': lambda x: 1.0 -
np.sum(np.abs(x))})
        else:
            bnds = tuple((0.0, 1.0) for x in range(n_assets))
            cons = ({'type': 'eq', 'fun': lambda x: 1.0 - np.sum(x)})

        opt_S = sco.minimize(
            loss,
            n_assets * [1.0 / n_assets],
            method='SLSQP', bounds=bnds,
            constraints=cons)

        optimal_weights = opt_S['x']

        # sometimes optimization fails with constraints, need to be
fixed by hands
        if self.allow_short:
            optimal_weights /= sum(np.abs(optimal_weights))
        else:
            optimal_weights += np.abs(np.min(optimal_weights))
            optimal_weights /= sum(optimal_weights)

        return optimal_weights


class MaxDecorrelationAgent:

    def __init__(
            self,
            portfolio_size,
            allow_short=False,
    ):

        self.portfolio_size = portfolio_size
        self.allow_short = allow_short
        self.input_shape = (portfolio_size, portfolio_size,)

    def act(self, returns):

        def loss(weights):
```

```
            weights = np.array(weights)
            return np.sqrt(np.dot(weights.T, np.dot(returns.corr(),
weights)))

        n_assets = len(returns.columns)

        if self.allow_short:
            bnds = tuple((-1.0, 1.0) for x in range(n_assets))
            cons = ({'type': 'eq', 'fun': lambda x: 1.0 -
np.sum(np.abs(x))})
        else:
            bnds = tuple((0.0, 1.0) for x in range(n_assets))
            cons = ({'type': 'eq', 'fun': lambda x: 1.0 - np.sum(x)})

        opt_S = sco.minimize(
            loss,
            n_assets * [1.0 / n_assets],
            method='SLSQP', bounds=bnds,
            constraints=cons)

        optimal_weights = opt_S['x']

        # sometimes optimization fails with constraints, need to be
fixed by hands
        if self.allow_short:
            optimal_weights /= sum(np.abs(optimal_weights))
        else:
            optimal_weights += np.abs(np.min(optimal_weights))
            optimal_weights /= sum(optimal_weights)

        return optimal_weights
```

3. Environment,py

```
import numpy as np
import pandas as pd

from utils import portfolio


class CryptoEnvironment:

    def __init__(self, prices = './data/crypto_portfolio.csv', capital =
1e6):
        self.prices = prices
        self.capital = capital
        self.data = self.load_data()

    def load_data(self):
        data =  pd.read_csv(self.prices)
        try:
            data.index = data['Date']
            data = data.drop(columns = ['Date'])
        except:
            data.index = data['date']
            data = data.drop(columns = ['date'])
        return data

    def preprocess_state(self, state):
        return state

    def get_state(self, t, lookback, is_cov_matrix = True,
is_raw_time_series = False):

        assert lookback <= t

        decision_making_state = self.data.iloc[t-lookback:t]
        decision_making_state =
decision_making_state.pct_change().dropna()

        if is_cov_matrix:
            x = decision_making_state.cov()
            return x
        else:
            if is_raw_time_series:
                decision_making_state = self.data.iloc[t-lookback:t]
            return self.preprocess_state(decision_making_state)

    def get_reward(self, action, action_t, reward_t, alpha = 0.01):

        def local_portfolio(returns, weights):
            weights = np.array(weights)
            rets = returns.mean() # * 252
            covs = returns.cov() # * 252
            P_ret = np.sum(rets * weights)
            P_vol = np.sqrt(np.dot(weights.T, np.dot(covs, weights)))
            P_sharpe = P_ret / P_vol
            return np.array([P_ret, P_vol, P_sharpe])

        data_period = self.data[action_t:reward_t]
        weights = action
        returns = data_period.pct_change().dropna()

        sharpe = local_portfolio(returns, weights)[-1]
```

```
        sharpe = np.array([sharpe] * len(self.data.columns))
        rew = (data_period.values[-1] - data_period.values[0]) /
data_period.values[0]

        return np.dot(returns, weights), rew



class ETFEnvironment:

    def __init__(self, volumes = './data/volumes.txt',
                       prices = './data/prices.txt',
                       returns = './data/returns.txt',
                       capital = 1e6):

        self.returns = returns
        self.prices = prices
        self.volumes = volumes
        self.capital = capital

        self.data = self.load_data()

    def load_data(self):
        volumes = np.genfromtxt(self.volumes, delimiter=',')[2:, 1:]
        prices = np.genfromtxt(self.prices, delimiter=',')[2:, 1:]
        returns=pd.read_csv(self.returns, index_col=0)
        assets=np.array(returns.columns)
        dates=np.array(returns.index)
        returns=returns.as_matrix()
        return pd.DataFrame(prices,
             columns = assets,
             index = dates
            )

    def preprocess_state(self, state):
        return state

    def get_state(self, t, lookback, is_cov_matrix = True,
is_raw_time_series = False):

        assert lookback <= t

        decision_making_state = self.data.iloc[t-lookback:t]
        decision_making_state =
decision_making_state.pct_change().dropna()

        if is_cov_matrix:
            x = decision_making_state.cov()
            return x
        else:
            if is_raw_time_series:
                decision_making_state = self.data.iloc[t-lookback:t]
            return self.preprocess_state(decision_making_state)

    def get_reward(self, action, action_t, reward_t):

        def local_portfolio(returns, weights):
            weights = np.array(weights)
            rets = returns.mean() # * 252
            covs = returns.cov() # * 252
            P_ret = np.sum(rets * weights)
            P_vol = np.sqrt(np.dot(weights.T, np.dot(covs, weights)))
            P_sharpe = P_ret / P_vol
```

```
            return np.array([P_ret, P_vol, P_sharpe])

        weights = action
        returns = self.data[action_t:reward_t].pct_change().dropna()

        rew = local_portfolio(returns, weights)[-1]
        rew = np.array([rew] * len(self.data.columns))

        return np.dot(returns, weights), rew
```

4. HRP_routines.py

```
import numpy as np
import pandas as pd
from scipy.cluster.hierarchy import dendrogram, linkage
from scipy.cluster.hierarchy import cophenet
from scipy.spatial.distance import pdist
import pylab



# On 20151227 by MLdP <lopezdeprado@lbl.gov>
# Hierarchical Risk Parity


def getIVP(cov, **kargs):
    # Compute the inverse-variance portfolio
    ivp = 1. / np.diag(cov)
    ivp /= ivp.sum()
    return ivp


def getClusterVar(cov,cItems):
    # Compute variance per cluster
    cov_=cov.loc[cItems,cItems] # matrix slice
    w_=getIVP(cov_).reshape(-1,1)
    cVar=np.dot(np.dot(w_.T,cov_),w_)[0,0]
    return cVar


def getQuasiDiag(link):
    # Sort clustered items by distance
    link = link.astype(int)
    sortIx = pd.Series([link[-1, 0], link[-1, 1]])
    numItems = link[-1, 3]  # number of original items
    while sortIx.max() >= numItems:
        sortIx.index = range(0, sortIx.shape[0] * 2, 2)  # make space
        df0 = sortIx[sortIx >= numItems]  # find clusters
        i = df0.index
        j = df0.values - numItems
        sortIx[i] = link[j, 0]  # item 1
        df0 = pd.Series(link[j, 1], index=i + 1)
        sortIx = sortIx.append(df0)  # item 2
        sortIx = sortIx.sort_index()  # re-sort
        sortIx.index = range(sortIx.shape[0])  # re-index
    return sortIx.tolist()


def getRecBipart(cov, sortIx):
    # Compute HRP alloc
    w = pd.Series(1, index=sortIx)
    cItems = [sortIx]  # initialize all items in one cluster
    while len(cItems) > 0:
        cItems = [i[j:k] for i in cItems for j, k in ((0, len(i) // 2),
(len(i) // 2, len(i))) if len(i) > 1]  # bi-section
        for i in range(0, len(cItems), 2):  # parse in pairs
            cItems0 = cItems[i]  # cluster 1
            cItems1 = cItems[i + 1]  # cluster 2
            cVar0 = getClusterVar(cov, cItems0)
            cVar1 = getClusterVar(cov, cItems1)
            alpha = 1 - cVar0 / (cVar0 + cVar1)
            w[cItems0] *= alpha  # weight 1
```

```python
            w[cItems1] *= 1 - alpha  # weight 2
    return w


def correlDist(corr):
    # A distance matrix based on correlation, where 0<=d[i,j]<=1
    # This is a proper distance metric
    dist = ((1 - corr) / 2.)**.5  # distance matrix
    return dist


def getHRP(cov, corr):
    # Construct a hierarchical portfolio
    dist = correlDist(corr)
    link = linkage(dist, 'single')
    #dn = sch.dendrogram(link, labels=cov.index.values,
label_rotation=90)
    #plt.show()
    sortIx = getQuasiDiag(link)
    sortIx = corr.index[sortIx].tolist()
    hrp = getRecBipart(cov, sortIx)
    return hrp.sort_index()
```

5. Stockmarket.py

```
6. # 1
import warnings

warnings.filterwarnings('ignore')

from keras.layers import Input, Dense, Flatten, Dropout
from keras.models import Model

import numpy as np
import pandas as pd
import os

import random
from collections import deque
import matplotlib.pylab as plt

from sklearn.decomposition import PCA

# 2
from environment import ETFEnvironment, CryptoEnvironment
from agent import MinVarianceAgent, MaxSharpeAgent,
MaxDecorrelationAgent, MaxReturnsAgent
from utils import *

# 3
N_ASSETS = 15  # 53
WINDOW_FIT = 180  # 252
WINDOW_HOLD = 90  # 252
env = CryptoEnvironment()  # ETFEnvironment

# 4
agent_max_returns = MaxReturnsAgent(N_ASSETS, allow_short=True)
agent_minvar = MinVarianceAgent(N_ASSETS, allow_short=True)
agent_maxsharpe = MaxSharpeAgent(N_ASSETS, allow_short=True)
agent_maxdecorr = MaxDecorrelationAgent(N_ASSETS, allow_short=True)

# 5
actions_equal, actions_returns, actions_minvar, actions_maxsharpe,
actions_maxdecorr = [], [], [], [], []
result_equal, result_returns, result_minvar, result_maxsharpe,
result_maxdecorr = [], [], [], [], []

for i in range(WINDOW_FIT, len(env.data), WINDOW_HOLD):
    state = env.get_state(i, WINDOW_FIT, is_cov_matrix=False)

    action_equal = np.ones(N_ASSETS) / N_ASSETS
    action_minvar = agent_minvar.act(state)
    action_max_returns = agent_max_returns.act(state)
    action_maxsharpe = agent_maxsharpe.act(state)
    action_maxdecorr = agent_maxdecorr.act(state)

    state_action = env.get_state(i + WINDOW_HOLD, WINDOW_HOLD,
is_cov_matrix=False)

    r = np.dot(state_action, action_equal)
    result_equal.append(r.tolist())
    actions_equal.append(action_equal)

    r = np.dot(state_action, action_minvar)
    result_minvar.append(r.tolist())
```

```
    actions_minvar.append(action_minvar)

    r = np.dot(state_action, action_max_returns)
    result_returns.append(r.tolist())
    actions_returns.append(action_max_returns)

    r = np.dot(state_action, action_maxsharpe)
    result_maxsharpe.append(r.tolist())
    actions_maxsharpe.append(action_maxsharpe)

    r = np.dot(state_action, action_maxdecorr)
    result_maxdecorr.append(r.tolist())
    actions_maxdecorr.append(action_maxdecorr)

# 6
result_equal_vis = [item for sublist in result_equal for item in
sublist]
result_returns_vis = [item for sublist in result_returns for item in
sublist]
result_minvar_vis = [item for sublist in result_minvar for item in
sublist]
result_maxsharpe_vis = [item for sublist in result_maxsharpe for item in
sublist]
result_maxdecorr_vis = [item for sublist in result_maxdecorr for item in
sublist]

# 7
plt.figure()
plt.plot(np.array(result_equal_vis).cumsum())
plt.plot(np.array(result_minvar_vis).cumsum())
plt.plot(np.array(result_returns_vis).cumsum())
plt.plot(np.array(result_maxsharpe_vis).cumsum())
plt.plot(np.array(result_maxdecorr_vis).cumsum())
plt.show()

# 8
print('EQUAL', print_stats(result_equal_vis, result_equal_vis))
print('MINVAR', print_stats(result_minvar_vis, result_equal_vis))
print('MAXRET', print_stats(result_returns_vis, result_equal_vis))
print('MAXSHRAPE', print_stats(result_maxsharpe_vis, result_equal_vis))
print('MAXDECORR', print_stats(result_maxdecorr_vis, result_equal_vis))

# 9
import matplotlib

current_cmap = matplotlib.cm.get_cmap()
current_cmap.set_bad(color='red')


# 10
def plot_results(benchmark_series,
                 target_series,
                 target_balances,
                 n_assets=N_ASSETS,
                 columns=state.columns,
                 name2plot='',
                 path2save='./',
                 base_name_series='series'):
    #     N = len(np.array(benchmark_series).cumsum())
    N = len(np.array([item for sublist in benchmark_series for item in
sublist]).cumsum())

    if not os.path.exists(path2save):
```

```
        os.makedirs(path2save)

    for i in range(0, len(target_balances)):
        current_range = np.arange(0, N)
        current_ts = np.zeros(N)
        current_ts2 = np.zeros(N)

        ts_benchmark = np.array([item for sublist in benchmark_series[:i
+ 1] for item in sublist]).cumsum()
        ts_target = np.array([item for sublist in target_series[:i + 1]
for item in sublist]).cumsum()

        t = len(ts_benchmark)
        current_ts[:t] = ts_benchmark
        current_ts2[:t] = ts_target

        current_ts[current_ts == 0] = ts_benchmark[-1]
        current_ts2[current_ts2 == 0] = ts_target[-1]

        plt.figure(figsize=(12, 10))

        plt.subplot(2, 1, 1)
        plt.bar(np.arange(n_assets), target_balances[i], color='grey')
        plt.xticks(np.arange(n_assets), columns, rotation='vertical')

        plt.subplot(2, 1, 2)
        plt.colormaps = current_cmap
        plt.plot(current_range[:t], current_ts[:t], color='black',
label='Benchmark')
        plt.plot(current_range[:t], current_ts2[:t], color='red',
label=name2plot)
        plt.plot(current_range[t:], current_ts[t:], ls='--', lw=.1,
color='black')
        plt.autoscale(False)
        plt.ylim([-1, 1])
        plt.legend()
        plt.savefig(path2save + base_name_series + str(i) + '.jpg')


# 11
plot_results(result_equal,
             result_maxdecorr,
             actions_maxdecorr,
             N_ASSETS,
             state.columns.tolist(),
             'Decorrelation portfolio', './images/decorr/', 'series')
# 12
plot_results(result_equal,
             result_maxsharpe,
             actions_maxsharpe,
             N_ASSETS,
             state.columns.tolist(),
             'Maximal Sharpe portfolio', './images/sharpe/', 'series')

# 13
plot_results(result_equal,
             result_minvar,
             actions_minvar,
             N_ASSETS,
             state.columns.tolist(),
             'Minimal variance portfolio', './images/minvar/', 'series')

# 14
```

```python
plot_results(result_equal,
             result_returns,
             actions_returns,
             N_ASSETS,
             state.columns.tolist(),
             'Maximal returns portfolio', './images/maxret/', 'series')
# 15
from agent import PCAAgent

# 50

agent_pca = PCAAgent(N_ASSETS, allow_short=True, pc_id=0)
actions_equal, actions_pca = [], []
result_equal, result_pca = [], []

for i in range(WINDOW_FIT, len(env.data), WINDOW_HOLD):
    state = env.get_state(i, WINDOW_FIT, is_cov_matrix=False)

    action_equal = np.ones(N_ASSETS) / N_ASSETS
    action_pca = agent_pca.act(state)

    state_action = env.get_state(i + WINDOW_HOLD, WINDOW_HOLD,
is_cov_matrix=False)

    r = np.dot(state_action, action_equal)
    result_equal.append(r.tolist())
    actions_equal.append(action_equal)

    r = np.dot(state_action, action_pca)
    result_pca.append(r.tolist())
    actions_pca.append(action_pca)

# 52
result_equal_vis = [item for sublist in result_equal for item in
sublist]
result_pca_vis = [item for sublist in result_pca for item in sublist]

# 53
plt.figure()
plt.plot(np.array(result_equal_vis).cumsum())
plt.plot(np.array(result_pca_vis).cumsum())
plt.show()

# 54
print('EQUAL', print_stats(result_equal_vis, result_equal_vis))
print('PCA', print_stats(result_pca_vis, result_equal_vis))



# 21
import matplotlib

current_cmap = matplotlib.cm.get_cmap()

# 22
plot_results(result_equal,
             result_pca,
             actions_pca,
             N_ASSETS,
             state.columns,
             'PCA PC0 portfolio', './images/pca/', 'series')

# 23
```

```python
from agent import HRPAgent

# 24
agent_hrp = HRPAgent(N_ASSETS, allow_short=True)

# 25
actions_equal, actions_hrp = [], []
result_equal, result_hrp = [], []

for i in range(WINDOW_FIT, len(env.data), WINDOW_HOLD):
    state = env.get_state(i, WINDOW_FIT, is_cov_matrix=False)

    action_equal = np.ones(N_ASSETS) / N_ASSETS
    action_hrp = agent_hrp.act(state)

    state_action = env.get_state(i + WINDOW_HOLD, WINDOW_HOLD,
is_cov_matrix=False)

    r = np.dot(state_action, action_equal)
    result_equal.append(r.tolist())
    actions_equal.append(action_equal)

    r = np.dot(state_action, action_hrp)
    result_hrp.append(r.tolist())
    actions_hrp.append(action_hrp)

# 26
result_equal_vis = [item for sublist in result_equal for item in
sublist]
result_hrp_vis = [item for sublist in result_hrp for item in sublist]

# 27
plt.figure()
plt.plot(np.array(result_equal_vis).cumsum())
plt.plot(np.array(result_hrp_vis).cumsum())
plt.show()

# 28
print('EQUAL', print_stats(result_equal_vis, result_equal_vis))
print('HRP', print_stats(result_hrp_vis, result_equal_vis))

# 29
import matplotlib

current_cmap = matplotlib.cm.get_cmap()

# 30
plot_results(result_equal,
             result_hrp,
             actions_hrp,
             N_ASSETS,
             state.columns,
             'HRP portfolio', './images/hrp/', 'series')

# 31
from agent import SmoothingAgent

agent_smooth = SmoothingAgent(N_ASSETS, allow_short=True,
forecast_horizon=WINDOW_HOLD)
actions_equal, actions_smooth = [], []
result_equal, result_smooth = [], []

for i in range(WINDOW_FIT, len(env.data), WINDOW_HOLD):
```

```
    state = env.get_state(i, WINDOW_FIT, is_cov_matrix=False,
is_raw_time_series=True)

    action_equal = np.ones(N_ASSETS) / N_ASSETS
    action_smooth = agent_smooth.act(state)

    state_action = env.get_state(i + WINDOW_HOLD, WINDOW_HOLD,
is_cov_matrix=False)

    r = np.dot(state_action, action_equal)
    result_equal.append(r.tolist())
    actions_equal.append(action_equal)

    r = np.dot(state_action, action_smooth)
    result_smooth.append(r.tolist())
    actions_smooth.append(action_smooth)

# 34
result_equal_vis = [item for sublist in result_equal for item in
sublist]
result_smooth_vis = [item for sublist in result_smooth for item in
sublist]

# 35
plt.figure()
plt.plot(np.array(result_equal_vis).cumsum())
plt.plot(np.array(result_smooth_vis).cumsum())
plt.show()

# 36
print('EQUAL', print_stats(result_equal_vis, result_equal_vis))
print('SMOOTHING', print_stats(result_smooth_vis, result_equal_vis))

# 37
plot_results(result_equal,
             result_smooth,
             actions_smooth,
             N_ASSETS,
             state.columns,
             'Holt smoothing portfolio', './images/smoothing/',
'series')

# 38
from agent import AutoencoderAgent

agent_ae = AutoencoderAgent(N_ASSETS, allow_short=True, encoding_dim=5)
actions_equal, actions_ae = [], []
result_equal, result_ae = [], []

for i in range(WINDOW_FIT, len(env.data), WINDOW_HOLD):
    state = env.get_state(i, WINDOW_FIT, is_cov_matrix=False,
is_raw_time_series=True)

    action_equal = np.ones(N_ASSETS) / N_ASSETS
    action_ae = agent_ae.act(state)

    state_action = env.get_state(i + WINDOW_HOLD, WINDOW_HOLD,
is_cov_matrix=False)

    r = np.dot(state_action, action_equal)
    result_equal.append(r.tolist())
    actions_equal.append(action_equal)
```

```
    r = np.dot(state_action, action_ae)
    result_ae.append(r.tolist())
    actions_ae.append(action_ae)

result_equal_vis = [item for sublist in result_equal for item in
sublist]
result_ae_vis = [item for sublist in result_ae for item in sublist]

plt.figure()
plt.plot(np.array(result_equal_vis).cumsum())
plt.plot(np.array(result_ae_vis).cumsum())
plt.show()

print('EQUAL', print_stats(result_equal_vis, result_equal_vis))
print('AUTOENCODER', print_stats(result_ae_vis, result_equal_vis))

plot_results(result_equal,
             result_ae,
             actions_ae,
             N_ASSETS,
             state.columns,
             'Autoencoder portfolio', './images/ae/', 'series')

# 45
import imageio
import glob

name = 'decorr'

filenames = sorted(glob.glob('./images/' + name + '/series*.jpg'))

filenames

images = []
for filename in filenames:
    images.append(imageio.imread(filename))
imageio.mimsave('./images/' + name + '_gif.gif', images, duration=0.5)
```

7. Stockmarketdata.csv

Date	RELIANCE	TCS	HINDUNILVR	HDFC Bank	INFOSYS	BHARTIARTL	KOTAKBKETF	ICICI Bank	I
########	894.38	2888.94	1337.86	1837.11	1135.86	394.41	248.09	262.9	
########	891.67	2907.92	1344.06	1827.19	1135.21	397.83	249.74	265.84	
########	901.57	2914.91	1352.9	1807.48	1131.81	399.14	249.39	268.92	
########	907.4	2949.56	1375.38	1820.17	1140.71	394.28	250.65	275.34	
########	912.81	2943.69	1371.58	1830.21	1128.81	386.64	252.29	279.81	
########	916.01	2936.88	1387.27	1839.27	1113.12	381.47	254.86	279.15	
########	919.12	2943.58	1392.16	1825.12	1115.77	379.78	255.88	289.29	
########	927.88	2987.68	1406.67	1813.54	1124.78	386.38	255.06	284.83	
########	929.39	3116.29	1406.2	1822.63	1157.71	381.34	255.37	287.38	
########	937.05	3181.37	1406.09	1835.33	1172.37	379.64	257.37	288.7	
########	932.57	3169.39	1414.59	1864.29	1125.66	377.4	258.04	287.29	
########	942.33	3165.81	1438.77	1879.58	1121.63	378.56	258.84	290.56	
########	941.25	3166.57	1449.94	1886.3	1124.27	386.56	257.54	291.81	
########	941.25	3190.56	1454.72	1867.66	1132.61	391.59	256.15	288.97	
########	928.21	3351.07	1455.55	1839.83	1171.78	397.85	252.98	284.43	
########	936.28	3478.05	1455.1	1835.94	1186.79	403.38	254.03	279.08	
########	959.86	3385.7	1454.35	1855.59	1168.64	402.57	254.91	284.41	
########	971	3445.24	1454.79	1847.36	1160.85	420.42	253.43	281.17	
########	977.14	3528.84	1479.86	1850.57	1172.76	411.68	253.47	277.94	
########	999.97	3475.49	1479.83	1852.06	1183.9	406.79	257.16	288.07	
########	969.39	3523.56	1500.69	1880.38	1199.78	409.6	260.13	285.71	
########	973.34	3496.99	1482.49	1896.68	1192.72	408.6	261.51	278.24	
########	964.27	3501.58	1451.68	1921.83	1195.66	403.77	261.31	281.51	
########	954.68	3485.14	1462.46	1914.82	1165.5	405.42	261.84	282.99	
########	963.82	3438.41	1493.07	1912.92	1181.67	400.03	262.81	287.62	
########	972.41	3447.8	1494.94	1905.65	1167.51	399.95	265.2	308.91	
########	973.59	3480.95	1498.62	1893.2	1169.95	403.14	265.05	307.23	
########	980.67	3478.59	1492.6	1896.16	1173.06	409.31	265.78	308.66	
########	986.84	3461.07	1498.41	1913.69	1174.3	389.19	267.01	309.93	
########	984.23	3427.81	1503.4	1927.77	1184.18	386.2	269.75	308.28	
########	988.75	3477.97	1524.08	1936.07	1196.6	382.08	270.75	311.45	
########	962.49	3494.21	1556.14	1921.18	1191.85	376.83	267.4	299.4	
########	949.68	3509.69	1581.28	1886.56	1186.19	371.49	265.94	297.56	
########	941.11	3487.86	1595.09	1858.85	1181.08	360.42	263.94	288.06	
########	927.05	3567.15	1593.71	1830.4	1182.21	364.41	263.27	290.11	

########	930.25	3520.43	1570.66	1808.53	1189.56	362.92	263.54	290.66
########	917.49	3513.84	1571.17	1791.87	1198.4	359.94	261.87	294.22
########	916.46	3579.69	1558.3	1802.61	1217.23	366.24	263.18	295.51
########	920.3	3633.11	1576.89	1812.9	1239.09	375.35	265.83	297.12
########	922.21	3516.38	1582.16	1828.99	1218.07	376.48	270.36	299.45
########	920.32	3517.42	1579.72	1817.47	1214.01	376.91	266.97	292.27
########	915.8	3508.9	1574.59	1793.23	1214.64	375.03	266.52	285.4
########	921.07	1742.44	1605	1818.76	1230.2	374.71	270.31	283.33
########	926.94	1742.29	1598.28	1844.43	1229.79	379.99	272.28	292.82
########	939.73	1730.25	1569.46	1845.64	1233.14	378.28	267.56	288.82
########	949.46	1728.88	1563.05	1835.68	1226.49	362.71	267.84	284.88
########	952.01	1725.85	1568.87	1850.8	1234.18	374.47	267.53	283.6
########	969.3	1738	1596.01	1868.34	1252.57	380.61	270.53	290.47
########	974.6	1748.59	1593.85	1841.82	1257.48	378.18	268.83	287.23
########	991.25	1752.55	1604.2	1842.44	1262.71	387.78	270.88	287.48
########	991.14	1773.52	1619.06	1847.89	1258.01	384.68	270.62	287.01
########	999.25	1818.17	1625.64	1836.34	1275.2	377.63	271.69	290.32
########	1001.22	1795.49	1608.43	1832.06	1242.78	378.11	271.85	286.05
########	1015.48	1823.42	1614.79	1834.12	1267.1	375.31	271.07	283.49
########	1014.97	1840.85	1619.1	1825.62	1273.18	375.78	270.18	290.76
########	1002.7	1829.83	1598.23	1827.21	1244.36	372.73	268.75	291.41
########	1015.65	1828.68	1607.29	1842.64	1245.07	373.45	268.73	293.04
########	1030.56	1822.83	1599.81	1858.55	1247.63	369.86	270.32	298.08
########	1014.6	1810.38	1603.17	1883.28	1249.94	371.81	269.68	298.78
########	1010.49	1817.52	1619.81	1894.14	1271.05	374.61	262.45	293.45
########	986.72	1842.52	1623.39	1913.98	1277.24	379.59	270.62	286.56
########	973.82	1870.51	1633.54	1917.81	1281.92	376.68	270.97	279.05
########	952.67	1846.23	1605.56	1880.17	1290.08	376.28	269.69	272.74
########	967.48	1851.25	1629.6	1895.06	1298.31	380.73	269.44	273.82
########	962.14	1852	1639	1895.71	1325.09	371.04	267.6	274.48
########	972.1	1872.16	1650.6	1881.84	1347.06	369.9	267.55	274.51
########	982.92	1863.61	1661.45	1909.61	1345.49	367.94	268.97	272.71
########	988.25	1874.33	1681.35	1926.73	1290.51	362.92	270.6	273.09
########	977.82	1907.65	1686.9	1925.99	1281.04	362.18	270.24	269.81
########	991.93	1888.15	1684.71	1908.8	1294.01	362.67	272.55	272.56
########	1018.86	1881.79	1687.42	1924.81	1305.09	365.71	273.79	274.11
########	1030.85	1951.99	1709.79	1944.19	1317.22	364.35	272.83	270.74
########	1080.29	1972.52	1739.12	1963.41	1299.24	362.91	275.56	271.72
########	1098.67	1986.23	1738.95	1970.08	1315.39	359.95	275.29	268.24
########	1082.43	1988.74	1762.14	1988.45	1345.93	350.92	273.77	259.58
########	1085.22	1994.6	1712.73	1983.3	1333.08	342.93	273.15	264.24
########	1091.64	2000.88	1655.5	2010.89	1325.16	336.43	276.43	263.58
########	1098.53	1984.27	1654.44	1989.14	1318.01	343.9	274.4	261.69

########	1125.34	1993.61	1653.2	1970.28	1352.12	344.58	274.15	265.82
########	1119.33	2005.83	1686.82	1971.96	1350.21	355.47	274.23	272.11
########	1115.46	1998.52	1679.97	1973.74	1368.92	361.67	276.58	274.97
########	1116.21	1979.25	1651.34	1998.75	1383.06	347.96	275.29	273.99
########	1113.9	1968.67	1661.25	2014.03	1377.36	356.44	279.46	283.67
########	1127.43	1948.53	1653.47	2036.67	1377.08	363.22	281.13	290.79
########	1143.36	1942.3	1682.66	2036.01	1356.75	374.32	282.65	302.62
########	1171.82	1934.85	1713.9	1987.35	1351.3	388.29	282.81	304.63
########	1193.58	1972.85	1731.56	1980.41	1355.85	391.75	282.99	301.76
########	1174.13	1960.41	1749.59	1942	1357.31	374.22	280.55	298.01
########	1175.98	1973.58	1756.65	1968.28	1361.34	373.61	282.28	304.11
########	1188.86	1981.25	1731.83	1979.47	1358.6	382.22	284.65	314.59
########	1186.35	1967.34	1730.71	1981.2	1366.36	382.15	285.19	314.62
########	1204.7	1969.57	1746.54	1975.61	1362.16	380.21	287.42	314.69
########	1221.41	1974.51	1753.38	1972.77	1381.59	369.03	288.27	334.73
########	1207.99	1985.67	1753.63	1963.96	1380.64	370.86	287.73	331.31
########	1191.83	1998.04	1743.98	1945.97	1401.07	371.52	285.09	324.83
########	1203.27	2010.34	1749.3	1953.53	1414.34	367.16	286.36	329.49
########	1204.69	2008.49	1740.41	1907.11	1426.82	370	285.13	335.63
########	1208.89	2012.85	1764.26	1892.02	1430.85	369.39	286.46	339.72
########	1227.54	2011.43	1790.26	1915.19	1389.04	370.78	289.15	339.52
########	1244.28	2015.38	1759.16	1916.42	1381.18	367.87	287.23	337.68
########	1259.12	2030.81	1769.65	1925.56	1392.99	370.67	287.93	336.89
########	1271.78	2041.86	1778.31	1914.46	1386.47	367.7	286.46	330.76
########	1290.31	2051.79	1789.67	1926.63	1406	376.01	288.1	338.81
########	1311.31	2065.5	1778.99	1946.42	1427.92	377.85	287.54	338.94
########	1309.99	2065.1	1759.56	1958.79	1412.92	377.96	289.91	340.57
########	1278.51	2078.06	1769.96	1938.39	1411.92	380.8	288.03	342.19
########	1247.15	2082.1	1778.75	1925.82	1439.05	380.95	287.02	342.19
########	1241.4	2072.5	1729.5	1939.87	1451.3	384.15	287.04	337.46
########	1241.44	2093.77	1645.16	1952.12	739.04	380.24	281.69	330.87
########	1230.06	2086.52	1617.43	1939.78	734.79	373.06	280.49	327.88
########	1245.01	2070.41	1615.22	1945.23	730.17	371.59	280.87	328.92
########	1272.05	2076.27	1629.77	1932.77	730.95	383.91	281.51	331.34
########	1263.51	2083.98	1613.62	1891.11	736.29	387.87	279.14	333.2
########	1248.31	2055.93	1589.84	1867.02	738.58	381.85	279.81	331.08
########	1243.62	2048.93	1619.28	1873.37	741.14	377.1	273.51	322.27
########	1258.4	2042.02	1623.81	1909.27	735.78	378.39	277.22	326.84
########	1232.59	2067.17	1605.71	1884.45	730.79	384.11	275.44	324.7
########	1228.49	2069.75	1654.67	1870.41	720.33	377.99	272.05	322.32
########	1214.62	2081.53	1658.44	1841.25	719.4	373.93	268.24	320.91
########	1223.31	2083.75	1625.59	1836.62	706.4	368.24	263.9	323.83
########	1225.23	2168.27	1605.21	1752.33	721.56	356.85	257.89	311.79

########	1226.67	2186.15	1615.46	1746.38	727.76	350.66	260.47	305.99
########	1239.18	2161.8	1610.72	1772.93	720.78	360.58	261.35	310.8
########	1252.61	2175.57	1617.39	1734.79	724.8	358.63	262.1	307.96
########	1259.94	2177.54	1611.94	1758.62	727.98	342.35	258.67	305.95
########	1231.68	2242.36	1627.99	1781.73	745.05	326.95	255	308.33
########	1218.04	2195.08	1625.03	1803.62	738.21	320.89	255.72	308.37
########	1140.4	2084.35	1580.24	1765.74	711.02	310.94	253.37	310.99
########	1079.24	2079.05	1550.21	1733.69	717.57	306.64	251.28	313.64
########	1082.04	2074.99	1558.57	1677.58	717.09	294.64	250.34	307.62
########	1105.24	2086.07	1522	1716.64	711.05	289.74	251.01	306.69
########	1101.57	2063.1	1517.7	1742.06	703.19	292.67	255.94	316.33
########	1086.1	1984.8	1499.63	1691.26	676.06	287.94	254.05	311.78
########	1121.38	1927.35	1560.56	1727.12	673.71	288.74	259.14	320.1
########	1136.88	1939.74	1521.84	1732.64	691.39	288.1	258.54	313.28
########	1154.23	1952.66	1543.19	1758.43	702.32	293.11	261.68	320.07
########	1166.52	1945.4	1564.12	1750.3	711.55	291.49	261.58	321.89
########	1092.05	1896.96	1570.69	1697.79	682.35	288.7	256.59	316.63
########	1084.3	1903.03	1582.54	1669.93	679.08	283.7	259.76	325.23
########	1053.27	1865.59	1562.87	1691.14	663.77	286.22	254.26	323.47
########	1050.33	1841.95	1572.51	1724.61	653.45	304.18	256.7	324.06
########	1029.24	1859.3	1569.01	1706.96	647.53	300.05	254.48	317.74
########	1043.01	1813.56	1566.91	1700.89	636.49	297.97	252.1	318.44
########	1069.59	1832.62	1538.93	1679.92	637.82	296.73	251.09	341.45
########	1065.23	1890.59	1581.09	1680.37	657.39	293.24	253.4	348.07
########	1059.51	1915.93	1603.21	1736.64	681.52	287.48	254.28	349.35
########	1059.03	1918.75	1597.61	1764.01	671.01	293.75	258.5	354.48
########	1072.3	1926.04	1635.23	1806.7	665.4	302.17	262.65	355.2
########	1081.82	1901.41	1626.11	1794.02	662.36	301.82	263.12	349.98
########	1104.23	1926.88	1634.23	1799.53	668.4	305.3	262.87	352.83
########	1110.53	1942.26	1642.77	1815.07	674.86	306.22	263.99	355.8
########	1101.88	1918.61	1666.33	1813.72	661.4	298.24	264.86	355.43
########	1085.52	1928.01	1667.56	1810.69	669.82	298.19	261.91	354.43
########	1090.11	1927.95	1659.66	1807.92	663.83	294.85	261.18	355.9
########	1098.36	1883.81	1717.8	1840.11	648.72	304.82	264.23	365.82
########	1095.12	1881.5	1710.7	1844.57	660.71	304.26	265.16	369.4
########	1121.66	1883.01	1695.82	1878.88	649.95	324.02	266.85	370.18
########	1142.86	1883.58	1693.34	1891.81	652.75	330.59	268.12	363.67
########	1143.18	1883.47	1699.17	1882.93	645.31	335.22	267.27	359.08
########	1116.81	1813.73	1690.74	1873.76	618.73	334.39	267.11	356.73
########	1113.39	1830.63	1683.93	1880.34	623.11	329.2	264.79	354.55
########	1107.7	1813.41	1727.56	1882.83	613.78	335.29	267.18	354.72
########	1121.98	1864.24	1725.98	1892.65	634.88	329.54	268.61	353.96
########	1144.01	1951.77	1732.38	1920.81	659.36	322.36	272.13	356.24

########	1170.72	1964.64	1763.68	1949.63	667.03	315.72	273.02	362.87
########	1171.45	1972.18	1763.32	1982.42	667.52	317.01	274.86	357.99
########	1158.74	1978.67	1794.19	1989.66	670.58	318.25	275.37	354.79
########	1149.79	2004.77	1817.21	1937.54	682.83	316.08	272.58	356.94
########	1150.35	2004.02	1826.46	1952.48	678.97	314.59	270.9	353.11
########	1136.17	1997.82	1815.81	1960.13	672.12	305.66	268.01	346.72
########	1124.39	1991.36	1811.81	1943.81	679.07	303.81	269.02	349.55
########	1091.18	1988.81	1802.24	1917.18	671.47	296.51	267.72	346.89
########	1079.85	1993.07	1795.87	1885.14	672.27	289.74	263.81	340.84
########	1099.51	2011.87	1826.8	1923.88	675.12	301.02	270.97	347.52
########	1110.87	1995.57	1853.19	1943.93	690.27	306.32	274.54	351.28
########	1107.1	1984.61	1852.52	1913.13	705.93	317.82	273	352.22
########	1125.03	1993.85	1855.7	1942.9	697.79	317.71	275.68	358.62
########	1129.68	1986.56	1839.35	1943.92	679.65	316.13	275.74	358.52
########	1141.63	1968.84	1848.62	1973.76	665.07	321.02	278.52	366.02
########	1125.05	1955.65	1839.16	1960.03	664.34	317.07	277.6	363.42
########	1109.5	1913.17	1816.56	1951.03	651.1	314.65	275.93	356.18
########	1094.03	1924.7	1797.63	1925.72	652.59	307.98	274.11	354.13
########	1081.5	1884.77	1772.77	1910.4	641.76	311.08	273.03	352.47
########	1119.69	1920.44	1807.59	1943.86	658.16	317.52	275.67	357.63
########	1129.79	1907.36	1826.37	1981.67	658.29	318.34	279.22	360.98
########	1122.29	1896.42	1820.78	1971.07	659.27	315.49	276.95	361.17
########	1119.01	1899.78	1804.57	1990.57	660.66	316.4	278.08	360.35
########	1113.42	1922.9	1786.89	1984.08	669.97	314.25	278.3	364.87
########	1101.37	1916.85	1792.38	1950.27	669.66	311.27	276.18	365.31
########	1093.06	1869.19	1784.94	1957.64	660.17	320.11	276.77	362.95
########	1110.32	1894.04	1790.16	1975.79	667.51	324.8	280.21	369.12
########	1103.82	1892.03	1773.74	1959.5	672.86	328.66	279.55	375.83
########	1110.55	1897.56	1785.48	1980.98	678.41	332.55	281.2	380.96
########	1106.3	1891.77	1787.93	1980.17	677.63	335.31	281.71	378.96
########	1099.31	1849.37	1773.87	1989.68	679.18	333.13	280.92	377.6
########	1093.55	1817.21	1766.64	1962.84	701.27	330.74	279.07	374.37
########	1124.77	1857.55	1780.97	1982.27	721.52	332.71	279.56	372.71
########	1139.05	1861.89	1774.6	1976.96	734.12	338.82	282.07	377.05
########	1139.3	1887.33	1760.88	1994.18	733.06	332.79	281.7	375.22
########	1162.97	1891.36	1746.76	2001.77	730.43	314.69	281.88	373.1
########	1222.77	1916.27	1747.7	2002.93	744.87	311.46	282.88	373.1
########	1234.02	1902.97	1749.37	1978.04	739.94	306.14	281.67	368.59
########	1233.79	1882.8	1774.32	1963.4	735.7	304.62	281.97	370.64
########	1240.42	1893.8	1763.94	1956.37	727.66	302.05	278.82	365.05
########	1253.45	1921.92	1763.2	1982.39	731.7	306.91	278.91	359.39
########	1237.32	1948.39	1751.66	1953.84	725.12	303.72	273.68	343.42
########	1212.83	1959.99	1750.93	1924.99	721.76	306.98	272	344.5

########	1208.06	1975.06	1740.74	1885.95	726.45	307.49	274.35	361.89
########	1220.45	2008.37	1764.22	1909.13	744.47	305.91	277.23	367.18
########	1240.68	2022.08	1802.24	1955.77	753.37	313.54	276.72	357.38
########	1269.56	2039.87	1807.77	1964.33	754.39	305.15	276.2	351.33
########	1292.06	2043.13	1825.73	1974.51	753.48	301.39	279.51	352.04
########	1309.33	2067.55	1830.56	1988.96	762.75	305.41	279.16	358.86
########	1302.19	2084.77	1839.14	1976.27	766.2	311.48	280.61	359.43
########	1282.97	2072.46	1827.11	1956.33	763.4	310.94	280.7	354.14
########	1259.38	2061.4	1804.64	1947.12	764.82	312.76	278.69	351.61
########	1259.38	2061.4	1804.64	1947.12	764.82	312.76	278.69	351.61
########	1261.8	2042.96	1796.79	1924.17	748.38	312.4	277.87	348.35
########	1258.94	2064.32	1803.74	1928.38	753.01	310.57	275.77	343.01
########	1227.02	2050.83	1786.78	1910.35	745.6	301.16	274.57	342.29
########	1228.62	2031.88	1765.01	1872.23	739.44	301.25	274.17	343.61
########	1226.57	2029.7	1746.17	1874.84	739.9	301.85	273.89	339.7
########	1226.57	2029.7	1746.17	1874.84	739.9	301.85	273.89	339.7
########	1228.49	1930.07	1741.74	1869.81	726.7	308.7	274.65	345.9
########	1228.29	1908.2	1731.28	1866.82	734.78	307.75	274.7	345.19
########	1246.51	1922.84	1750.18	1881.67	732.23	306.1	276.05	352.18
########	1233.91	1920.72	1765.88	1888.58	733.71	313.75	275.21	352.44
########	1232.29	1925.65	1767.01	1886.54	749.2	316.43	276.24	355.64
########	1232.29	1925.65	1767.01	1886.54	749.2	316.43	276.24	355.64
########	1221.22	2022.12	1776.55	1865.27	745.77	316.66	276.73	349.4
########	1227.12	2050.25	1745.45	1845.85	738.56	319.12	275.28	347.94
########	1233.07	2009.42	1739.45	1841.82	735.59	320.07	275.32	347.9
########	1229.08	1993.73	1739.39	1851.62	740	307.61	277.39	352.79
########	1229.02	1987.11	1724.93	1857.2	734.79	306.51	279.03	358.98
########	1257.64	2001.3	1707.44	1873.84	732.02	310.52	282.94	370.13
########	1271.7	2014.4	1708.79	1878.9	724.84	309.38	284.08	370.8
########	1267.28	2023.03	1702.7	1879.97	714.77	309.59	283.88	370.45
########	1291.98	2017.1	1716.43	1895.34	712.07	325.7	286.46	375.79
########	1324.88	2015.6	1734.4	1915.08	708.99	346.61	290.64	386.34
########	1339.1	2001.12	1747.33	1946.93	710.81	343.85	292.03	389.13
########	1352.38	1991.28	1736.25	1959.13	710.02	340.45	296.16	389.17
########	1334.04	2040.01	1702.04	1972.9	717.57	336.67	299.31	394.71
########	1341.23	2033.02	1698.64	1968.65	715	333.42	301.2	396.52
########	1362.21	2010.71	1698.3	1968.78	715.22	336.06	302.81	398.15
########	1376.17	2025.82	1702.74	1984.87	737.53	334.89	304.34	394.49
########	1375.45	2015.05	1685.25	1991.26	741.4	337.63	304.26	392.9
########	1375.45	2015.05	1685.25	1991.26	741.4	337.63	304.26	392.9
########	1325.16	1987.88	1674.08	1952.32	735.24	328.45	298.32	384.41
########	1353.06	1971.46	1683.21	1944.21	724.65	328.27	301.91	389.46
########	1360.73	1979.12	1693.98	1937.71	730.32	327.92	307.34	394.23

########	1357.34	1993.96	1681.5	1942.93	739.73	327.25	308.43	398.93
########	1363.56	2001.7	1706.33	1956.38	743.49	332.89	316.02	399.11
########	1386.73	2027.91	1702.01	1960.06	754.32	340.15	310.68	404.53
########	1390.35	2064.13	1687.2	1979.73	760.56	353.2	309.65	395.93
########	1390.2	2073.86	1677.49	2019.73	752.48	357.37	306.38	397.01
########	1360.64	2039.77	1671.46	2037.46	748.43	356.35	308.16	391
########	1352.52	2037.19	1656.32	2055.28	758.69	356.24	305.17	388.82
########	1335.09	2056.64	1669.4	2048.64	762.8	357.93	307.41	388.41
########	1330.46	2085.63	1673.99	2057.72	765.16	348.74	306.45	392.6
########	1337.99	2053.91	1694.92	2038.39	759.53	347.35	307.32	395.63
########	1338.41	2028.47	1710.78	2029.46	744.27	345.92	303.85	390.26
########	1346.08	2019.97	1718.84	2020.07	746.82	343.75	305.57	392.26
########	1341.32	2086.02	1722.9	2019.1	724.58	341.45	307.58	394.89
########	1349.7	2137.54	1736.15	2025.8	722.4	346.14	310.61	404.8
########	1379.51	2138.81	1738.82	2013.04	716.8	341.94	310.2	406.69
########	1353.82	2165.74	1738.7	1970.41	721.04	346.74	304.06	398.24
########	1362.98	2147.21	1742.1	1944.34	726.32	319.44	302.72	397.18
########	1379.35	2172.32	1742.32	1958.44	733.43	322.13	302.19	397.78
########	1391.77	2189.55	1743.94	1970.35	730.72	327.66	305.01	398.27
########	1383.98	2225.36	1738.85	1963.96	736.11	327	304.39	402.32
########	1384.81	2252.48	1755.94	1976.43	748.61	318.75	302.22	407.21
########	1402.33	2227.88	1740.05	2012.44	738.19	326.15	303.02	399.85
########	1412.19	2153	1698.94	2020.47	725.26	330.73	305.18	400.66
########	1392.39	2150.11	1666.58	1967.93	718.07	332.08	302.49	400.54
########	1361.64	2163.53	1700.33	1983.01	724.51	330.53	301.48	397.17
########	1313.1	2151.85	1686.13	1941.55	721.65	323.94	297.29	384.93
########	1265.1	2166.98	1695.24	1918.15	724.48	320.52	294.65	382.09
########	1263.56	2161.83	1694.39	1936.05	717.88	322.96	295.61	385.98
########	1246.96	2143.87	1705.6	1958.85	720.09	323.43	294.03	380.82
########	1252.66	2101.14	1696.91	1951.75	711.18	330.06	294.42	382.41
########	1269.53	2103.66	1681.14	1961.65	719.58	333.58	294.87	381.46
########	1264.7	2107.55	1686.43	1935.38	730.03	326.23	294.73	380.92
########	1268.85	2107.34	1729.2	1980.19	725.44	326.01	298.17	386.58
########	1319.23	2119.5	1761.28	2093.43	719.31	335.84	310.75	409.17
########	1348	2122.37	1786.16	2145.11	712.09	334.15	310.28	404.16
########	1345.18	2091.31	1761.42	2138.55	712.73	334.03	310.94	406.02
########	1364.45	2071.74	1775	2168.72	708.67	341.53	317.58	417.39
########	1334.73	2051.14	1744.24	2120.74	706.58	349.09	315.74	424.41
########	1322.83	2054.87	1757.68	2161.74	711.8	349.01	321.34	433.75
########	1323.88	2076.46	1773.19	2140.11	725.58	345.44	322.01	433.06
########	1321.09	2109.46	1778.64	2150.97	729.38	340.4	321.01	425.79
########	1329.55	2142.62	1786.56	2169.63	733.51	346.13	322.4	424.71
########	1330.68	2184.11	1784.89	2180.23	737.91	348.68	324.14	424.04

########	1344.47	2223.68	1817.64	2217.73	742.27	351.93	322.87	423.15
########	1358.17	2195.01	1817.55	2222.85	736.49	352.48	324.24	420
########	1335.47	2169.63	1838.66	2200.67	736.86	352.01	317.29	414.37
########	1314.55	2161.21	1830.55	2191.61	739.65	357.2	316.89	414.74
########	1319.5	2219.98	1852.7	2193.98	751.37	360.22	318.28	417.03
########	1327.51	2250.89	1842.68	2192.02	754.71	362.23	318.69	419.85
########	1332.57	2241.29	1837.61	2174.02	750.69	359.89	318.09	417.27
########	1322.47	2265.77	1836.77	2182.23	743.56	362.51	316.21	418.62
########	1316.01	2250.08	1831.03	2189.8	741.51	355.79	313.64	418.86
########	1291.9	2252.43	1816.94	2173.16	741.8	348.21	310.76	415.63
########	1279.9	2245.75	1802.57	2165.07	749.2	343.69	311.14	421.58
########	1284.31	2262.19	1813.27	2173.69	751.2	341.76	312.94	422.34
########	1288.38	2248.36	1801.08	2193.71	749.22	347.04	312.39	427.74
########	1282.04	2255.87	1774.8	2153.08	750.28	345.62	313.4	431.95
########	1264.03	2267.8	1764.09	2148.03	747.99	346.24	313.34	430.92
########	1281.36	2264.08	1754.21	2160.66	744.67	347.21	313.96	429.37
########	1295.88	2259.22	1756.61	2165.85	743.56	346.87	317.32	437.59
########	1281.11	2254.5	1777.67	2189.62	733.31	346.63	320.45	441.05
########	1261.89	2234.73	1782.4	2192.12	732.85	346.01	319.78	437.24
########	1263.04	2237.15	1783.43	2227.98	733.32	347.66	321.57	440.52
########	1272.92	2237.79	1784.46	2262.05	734.99	351.74	320.65	436.41
########	1281.62	2242.16	1785.7	2275.44	736.8	354.15	322.11	436.58
########	1285.9	2241.04	1788.87	2275.11	733.38	361.71	324.35	438.41
########	1275.14	2179.39	1804.5	2284.65	723.71	365.11	322.24	436.79
########	1256.53	2168.49	1758.08	2259.59	717.2	359.53	316.99	427.7
########	1272.67	2128.96	1741.37	2220.71	713.79	359.07	312.81	427.87
########	1277.53	2102.52	1733.53	2232.59	716.16	357.83	313.45	430.21
########	1284.68	2103.2	1731.52	2254.42	716.61	359.8	313.94	426.97
########	1292.37	2108.31	1724.02	2271.09	724.08	355.91	315.57	427.77
########	1276.79	2126.77	1715.21	2259.05	769.12	352.66	312.14	423.79
########	1285.92	2108.89	1740.12	2272.13	782.56	351.32	313	427.13
########	1287.12	2111.55	1751.21	2284.73	789.65	348.45	314.51	424.69
########	1269.25	2084.29	1735.42	2332.67	793.85	345.25	313.99	421.3
########	1256.85	2080.79	1733.33	2313.11	786.53	340.26	310.09	413.12
########	1262.08	2099.33	1684.64	2201.94	787.13	340.71	301.2	409.91
########	1281.65	2120.71	1688.7	2143.22	795.33	343.02	300.64	413.88
########	1261.14	2097.5	1708.56	2175.12	786.78	338.41	298.04	406.86
########	1245.25	2118.42	1738.55	2196.99	793.82	340.16	300.22	409
########	1222.79	2109.59	1729.53	2178.06	786.88	333.22	300.85	413.72
########	1212.37	2123.42	1715.42	2142.3	790.84	333.96	300.88	430.36
########	1199.24	2162.72	1719.38	2132.05	793.92	344.41	300.11	432.02
########	1173.86	2180.17	1730.76	2119.57	792.38	340.98	295.24	425.24
########	1163.85	2195.1	1730.7	2083.17	774.33	329.6	292.37	419.1

########	1181.63	2185.91	1729.04	2097.26	770.48	337.94	287.66	410.35
########	1143.81	2241.19	1739.21	2132.6	777.83	348.5	282.15	401.22
########	1137.27	2232.27	1745.62	2179.27	776.57	365.55	286.04	408.32
########	1118.77	2223.14	1775.25	2178.88	778.99	368.43	285.54	411.55
########	1124.86	2247.44	1795.8	2167.66	787.81	373.28	285.29	409.14
########	1166.33	2253.09	1829.51	2222.52	790.13	369.63	292.71	419.14
########	1274.8	2219.61	1824.91	2136.79	772.53	352.5	287.52	415.92
########	1292.12	2200.61	1833.58	2125.76	773.33	359.15	285.89	418.24
########	1280.75	2173.77	1832.3	2114.45	772.37	360.31	286.95	416.36
########	1289.52	2171.81	1827.1	2118.78	779.02	363.19	289.76	424.67
########	1280.83	2190.28	1836.93	2099.65	793.28	359.61	285.99	417.39
########	1271.42	2193.43	1850.21	2079.25	799.47	356.83	285.56	415.25
########	1251.88	2213.54	1871.88	2034.79	797	354.18	281.67	404.13
########	1258.18	2244.15	1862.24	2019.52	802.51	355.35	278.23	394.27
########	1274.2	2259.83	1867.97	2112.51	799.09	358.35	287.91	404.34
########	1272.49	2241.51	1862.39	2171.77	785.23	349.6	289.93	417.06
########	1266.27	2233.18	1828.31	2177.34	796.96	344.52	285.83	414.46
########	1247.45	2235.21	1831.49	2140.01	805.11	344.75	282.44	404.32
########	1237.5	2241.92	1859.34	2149.86	809.24	344.46	280.79	407.42
########	1222.13	2275.88	1852.01	2100.34	816.96	340.35	277.52	396.27
########	1196	2248.41	1836.71	2095.5	818.11	343.06	279.04	395.72
########	1202.7	2225.26	1827.43	2054.36	833	345.5	277.44	390.76
########	1215.37	2204.18	1824.47	2037.08	840.49	350.02	278.36	392
########	1224.14	2188.09	1843.73	2060.07	830.92	353.46	279.31	396.17
########	1233.81	2162.64	1830.91	2070.62	818.82	355.81	283.31	395.79
########	1222.41	2135.46	1818.67	2096.43	816.08	349.7	285.88	404.39
########	1216.36	2135.83	1810.25	2075.39	824.18	343.05	285.71	407.77
########	1205.64	2151.17	1816.87	2052.4	827.38	343.12	286.65	412.22
########	1201.83	2123.78	1830.88	2006.18	827.54	337.97	282.89	406.05
########	1205.99	2133.02	1827.69	1993.65	828.68	336.72	279.58	400.63
########	1189.5	2103.74	1827.22	1985.51	820.99	337.55	275.55	388.74
########	1238.4	2083.7	1925.76	2038.33	811.86	352.45	293.41	411.8
########	1249.95	2011.62	2066.01	2146.38	770.66	352.36	309.94	441.55
########	1282.47	2045.07	2041.07	2138.91	785.94	348.86	311.34	439.43
########	1284.48	2077.98	2030	2071.36	793.69	342.67	304.97	434.35
########	1293.25	2084.45	2034.46	2068.89	783.93	344.13	308.6	449.49
########	1307.46	2065.56	2017.61	2061.64	785.46	345.53	308.76	447.4
########	1322.09	2090.07	1991.92	1989.56	801.63	356.16	304.88	436.05
########	1320.17	2059.04	1999.2	1994.56	796.66	356.45	300.24	431.03

Investment and portfolio management

ORIGINALITY REPORT

5%	3%	1%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijpam.eu Internet Source	2%
2	Ye Wang, Yanju Chen, YanKui Liu. "Modeling Portfolio Optimization Problem by Probability-Credibility Equilibrium Risk Criterion", Mathematical Problems in Engineering, 2016 Publication	1%
3	Submitted to National Institute of Technology Delhi Student Paper	1%
4	Submitted to Kenyatta University Student Paper	<1%
5	Submitted to Coventry University Student Paper	<1%
6	Submitted to Amity University Student Paper	<1%
7	Submitted to University of East London Student Paper	<1%
	Submitted to Nizwa College of Technology	

8	Student Paper	<1%
9	scikit-learn.org Internet Source	<1%
10	Submitted to University of Northumbria at Newcastle Student Paper	<1%
11	Submitted to B.S.Abdur Rahman Crescent Institute of Science & Technology Student Paper	<1%
12	export.arxiv.org Internet Source	<1%
13	Submitted to National College of Ireland Student Paper	<1%
14	www.edureka.co Internet Source	<1%
15	Submitted to PSB Academy (ACP eSolutions) Student Paper	<1%
16	Submitted to University of Lancaster Student Paper	<1%
17	Submitted to Imperial College of Science, Technology and Medicine Student Paper	<1%
18	Submitted to International School of	<1%

	Management Student Paper	
19	Submitted to University of Hull Student Paper	<1%
20	www.essaybaywriters.com Internet Source	<1%
21	Submitted to South Bank University Student Paper	<1%
22	Submitted to Management Development Institute Of Singapore Student Paper	<1%
23	Submitted to College of Banking and Financial Studies Student Paper	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		