

# Conversational User Interface for Personal Wealth Management

by

Naga Raja Paidimarri

A report submitted in conformity with the requirements  
for the Master of Engineering Project  
Mechanical and Industrial Engineering  
University of Toronto

© Copyright by Naga Raja Paidimarri (2019)

# Conversational User Interface for Personal Wealth Management

Naga Raja Paidimarri

Master of Engineering in Mechanical and Industrial Engineering

Mechanical and Industrial Engineering  
University of Toronto

2019

## Abstract

Machine learning (ML) is disrupting diverse fields and finance is no exception. Recently, many financial services companies have started to incorporate machine learning into their traditional work. Some of the major application of ML in finance are Algorithmic trading, sentiment analysis, fraud detection, enhancing customer service either through conversational user interfaces or providing product recommendations and underwriting. Today more companies are using conversational user interfaces to reduce costs (manpower) and improve their customer service and in this project the potential advantages of using such user interfaces for personal investment management is explored. With this motivation in mind, a prototype of the chatbot was developed. This chatbot can understand investors preferences such as active or passive strategy and create portfolios. The bot will be easy and intuitive to use, and it will guide users throughout the conversation.

## Acknowledgments

First, I would like to thank Professor Roy Kwon for accepting to supervise this project and Professor Oleksandr Romanko for giving me the opportunity to work on an exciting project. This project would not have been possible without the continuous support of professor Romanko and I am thankful for his patience, motivation, enthusiasm and immense knowledge.

# Table of Contents

## Contents

Acknowledgments.....	iii
Table of Contents.....	iv
1 Introduction.....	6
2 Wealth Management and Risk Profiling.....	7
3 Conversational User Interface.....	8
3.1 Advantages of using chatbots .....	9
3.2 Chatbot for investment.....	11
3.3 Platforms .....	12
4 IBM Watson .....	13
4.1 Why IBM Watson .....	13
4.2 IBM Watson Assistant key features.....	13
4.3 Limitations .....	14
5 Methodology .....	15
5.1 How Chabot works .....	15
5.1.1 Intents.....	15
5.1.2 Entities .....	15
5.1.3 Dialog Flow .....	15
5.2 Flow diagram of the bot.....	16
5.3 Python .....	17
5.4 Active Investing – Portfolio Optimization.....	18
5.4.1 Modern Portfolio Theory .....	18
5.4.2 Mean-Variance Portfolio Optimization .....	19
5.5 Passive Investing.....	24

6	Limitations of Chatbot .....	25
7	Chatbot Testing .....	26
8	Conclusions .....	27
9	Future Work/ Recommendations .....	28
9.1	Technical Improvements: Optimization Formulation.....	28
9.2	Functional Improvements: .....	29
9.3	Deploying the Bot .....	29
10	Appendix .....	30
10.1	Active strategy conversation .....	30
10.2	Passive strategy conversation .....	32
10.3	Functions .....	33

# 1 Introduction

The report is structured in the following way:

*Wealth management and risk profiling:* In this section wealth management and the importance of risk profiling are explained. Next, the problems with current risk profiling practices in specific, using questionnaires for determining risk preferences is discussed. Later, the methodology I adapted in the bot is explained along with potential advantages of it.

*Conversational User Interface:* For any project understanding the commercial applications is important so in this section why companies are adapting chatbots and their advantages are discussed. One may assume that people use chatbots out of necessity, but it is empirically found out this is not the case and customers also see value in using chatbots. Now after understanding the advantage of chatbots and convinced that people would use them if properly designed the need of using automated advising in wealth management is discussed. This is followed by the advantages of using chatbot in investing and the inherent advantage of using chatbots over robo advisors.

*IBM Watson:* The prototype has been developed in the python and for NLP IBM Watson assistant has been used. The advantages of using IBM Watson and some of its key features are mentioned. It is not easy to choose the right platform out of many vendors available and some of the blogs and are referenced to guide others.

*Methodology:* The basic building blocks of a conversational assistant are discussed along with examples used. A flow chart of the prototype is presented and a simple conversation with the bot is discussed in detail. Next, an attempt has been to explain different portfolio optimization techniques and the focus was not explain these techniques in detail but to understand how the optimization problem is formulated in practice. Portfolio optimization strategies include mean variance optimization both for minimizing absolute risk and risk relative to a bench mark, robust optimization, CVaR optimization and Risk parity.

In the following sections some of the limitations of the chatbot and important metrics to test them are provided. Finally, after conclusion, in the future work section I explained the further improvement to the bot I will do in future.

## 2 Wealth Management and Risk Profiling

Wealth management is a broad term and it generally encompasses investment management with other financial services such as legal or estate planning, accounting and tax services to manage the financial situation of an investor. Investment management is more specific, and investment managers/financial advisors are more concerned about investing in financial assets after understanding the needs and preferences of the investors.

One of the important steps in wealth/investment management is Risk profiling. Risk profiling is an umbrella term that is used to describe the various facts and traits about the investor that needs to be considered before recommending a portfolio. Unless the investors risk preferences, time horizons, liquidity needs, and goals are correctly determined one cannot recommend suitable portfolios. Despite the importance of risk profiling, what constitutes an accurate and adequate risk profile is not completely defined. To be specific, which facts and traits of the client are to be included in the risk profile is often subjective. Adding to the uncertainty, *the current practice of risk profiling through questionnaires is found empirically to be unreliable and explains only 15% of variations in portfolios of investors.*<sup>1</sup> The major culprit is in the design of questionnaires and they generally miss the other factors which are equally crucial for risk profile such as, the financial decisions the investor made in the past and the influence of his social environment.

This report by CFA institute<sup>1</sup> also contains more information on the studies corresponding to risk profiling and further divided the risk in to ‘risk capacity’ and ‘risk aversion’. Risk capacity corresponds to present economic conditions and risk aversion represents the psychological traits of an investor. The report also found in the questionnaire’s they studied are not robust for instance, a loss of 1000\$ would be devastating for a student but implies nothing for a working professional. A more appropriate method would be to use percentages and values more representative to the context of the investor. This is necessary to generate accurate emotional reaction the financial loss would cause. The report concludes by providing some practical guidelines but still falls short on completely describing them and one such guideline is “to collect information about the investor's parents investing attitudes “ and this may not be comfortable for everyone to share.

---

<sup>1</sup> Klement, J. (2015). *INVESTOR RISK PROFILING: AN OVERVIEW*(Rep.). doi:ISBN 978-1-934667-93-4

The report, in the end, added that “not all investors like to answer generic questionnaires and maybe some like to tell stories”. This indicates that a personalized approach for each investor could be a better approach for risk profiling.

Finally, it can be inferred that risk profiling is more than asking a set of questions and there is no one definite way to determine it accurately. *For this project, I adapted “results first approach”.* *The major advantage of this approach is, it is easy to implement.* The problem of asking questionnaire is completely avoided and instead, five portfolios with different risks are selected across the efficient frontier and are presented to the investor. When implemented in a chatbot the investor can try and see different combinations all by entering a simple conversation. The chatbots have the potential to make this process efficient and productive and by changing the design we can urge investor to simulate different portfolios. Before discussing more, I would like to talk about chatbots first.

### 3 Conversational User Interface

The conversational user interface or conversational assistant or more simply chatbot is a piece of a computer program designed with an aim to maintain conversations with humans in natural language.<sup>2 3 4</sup> Artificial Intelligence and Natural language processing are the two basic building blocks which thrived innovation in chatbots from being a mere pattern based (matching of keywords) to more advanced bots capable of understanding context (natural language understanding) and retrieving relevant information. Some of the most common examples of a chatbot with voice interface are Apple’s ‘Siri’ and Amazon’s ‘Alexa’

Chatbot phenomena are not very new and the first known chatbot was Eliza (developed in 1966). Its goal was to behave as a Rogerian psychologist<sup>5</sup>. ELIZA used a simple pattern or keyword matching to provide response irrespective of the context (as long as specific words are present it

---

<sup>2</sup> Jyothirmayi1, N., Soniya, A., Grace, Y., C, K., & B.V, R. (2019). SURVEY ON CHATBOT CONVERSATIONAL SYSTEM. : *Journal of Applied Science and Computations*,6(1), 285-297. doi:ISSN NO: 1076-5131

<sup>3</sup> Shawar, Bayan & Atwell, Eric. (2007). *Chatbots: Are they Really Useful?*. LDV Forum. 22. 29-49.

<sup>4</sup> Zumstein, Darius & Hundertmark, Sophie. (2017). *Chatbots – An Interactive Technology for Personalized Communication, Transactions and Services*. IADIS International Journal on WWW/Internet. 15. 96-109.



would give the same response) the words are used in. Chatbots have evolved over time and they are tabulated<sup>5</sup> in many papers.

Consulting companies tend to classify chatbots according to the task they perform, and academic literature may classify them based on the algorithms and methodology used. There is a good amount of literature available on both. As of 2019, all major technology companies such as Google, Microsoft, Amazon and Facebook offer platforms to build conversational assistants with the minimum code as possible.

### 3.1 Advantages of using chatbots

The recent advancement in AI made chatbots more robust and suitable for a wide range of applications. Using chatbots companies can continue to serve their customers 24 hours every day and be accessible to them independent of work hours. With chatbots (a type of automation) companies can rapidly scale up and down as needed depending on the demand.

Presently companies around the world are helping their customers by providing chatbots for searching & booking trips, news and weather information, reserving or purchasing tickets, buying products on e-commerce websites. Presently there are more than 100'000 chatbots only in Facebook messenger and the potential global annual revenue generated by chatbot transactions is estimated up to 32 billion USD. Chatbots apart from boosting revenues they also reduce costs. In the US Insurance industry alone, salary savings by chatbots are estimated to 12 billion USD. Similarly, 15 billion USD in financial services and 23 billion USD in customer service<sup>6</sup>.

The advantages to companies are clear but will the customers be willing to accept this change and the answer is yes. This empirical study<sup>7</sup> tried to find out what is the motivation for customers to use chatbots. *Majority of participants (68%) reported productivity to be the main reason for using chatbots.* To be specific as expected, the participants highlighted the ease, speed, and convenience

---

<sup>5</sup> Bradesko, L., & Mladenec, D. (2012). A Survey of Chabot Systems through a Loebner Prize Competition.

<sup>6</sup> Zumstein, D., & Hundertmark, S. (2017). CHATBOTS – AN INTERACTIVE TECHNOLOGY FOR PERSONALIZED COMMUNICATION, TRANSACTIONS AND SERVICES. *IADIS International Journal*, 15(1), 96-109. Retrieved April 21, 2019, from <http://www.iadisportal.org/ijwi/>

<sup>7</sup> Brandtzaeg, P. B., & Følstad, A. (2017). Why People Use Chatbots. *Internet Science Lecture Notes in Computer Science*, 377-392. doi:10.1007/978-3-319-70284-1\_30

of using chatbots. The chatbots are easy to adapt and they provide information quickly and they are convenient because they are accessible all the time. These factors make chatbots as the most efficient platform for information provided, they are properly designed. Participants also acknowledge that chatbots provide assistant and access to information and they improve productivity.

Now what if the customers are using chatbots because they had to but contrary to the general assumption that people might feel less confident or uncomfortable talking to an artificial program trying to mimic human conversation the study in this paper<sup>8</sup> concludes that people are actually inclined to send more messages to chatbots and the reason they said is people try to model their communication to match the chatbots. So, if properly designed and being upfront about what our chatbots can offer it is has good potential to work<sup>8</sup>.

It is not a simple and straight forward exercise for any company and more so financial services industries to replace their financial advisors with a chatbot or provide a hybrid environment of human and chatbots to their existing customers. The authors in this paper<sup>9</sup> point to a technology company whose newly built chatbot adoption rates were less than expected and the primary reason for this is most chatbots fail to understand user's needs. So, before a financial company builds a chatbot they should understand user's motivation and develop bots which have clear purposes (only helping with portfolio building) and usability (giving users enough tools to make a decision).

---

<sup>8</sup> Hill, J., Ford, W. R., & Farreras, I. G. (2015). Real conversations with artificial intelligence: A comparison between human-human online conversations and human-chatbot conversations. *Computers in Human Behavior*, 49, 245-250. doi:10.1016/j.chb.2015.02.026

<sup>9</sup> Brandtzaeg, P. B., & Følstad, A. (2017). Why People Use Chatbots. *Internet Science Lecture Notes in Computer Science*, 377-392. doi:10.1007/978-3-319-70284-1\_30

### 3.2 Chatbot for investment

Technology companies like Amazon and Facebook have implemented AI technologies since some time and recently financial services industries started using them. Some banks are reported <sup>10</sup> to be using AI in areas of marketing, risk management and for preventing frauds. AI has made these processes more efficient and effective by personalizing to each user, reducing the costs by automating, for instance, customer service. Some companies have started seeing the potential of AI in wealth management and one such phenomenon which surged recently and is a crude form of a chatbot are Robo advisors.

Financial advising is not accessible for all investors equally because of high costs associated and the uncertain nature of the market and these factors paved the way for automated advice where the algorithms provide financial advice, manage and rebalance portfolios. The other key features of robo advisors are low fees, most efficient algorithms are being accessible to every investor and one can start investing a small amount, the first robo advisor “Betterment” was started immediately after the recent financial crisis<sup>11</sup>. Robo advisors are projected to reach \$2.2 Tn by 2020.

The question is now set for why someone should use a chatbot for investment management. After all, adding a nice user interface to robo advisors could do the same thing but, I would argue that replacing robo advisors with chatbots can add more value to the users. Robo advisors presently, are designed to require investors to answer questions to capture information. *They lack the interaction, personalization and most often investors must rely on FAQ's and send an email to supporting team waiting for some time to get answers.* A potential advice for this problem is to use chatbot based robo advisors. Chatbots are evidently the solutions to these problems and it has all the advantages of robo advisor inherently along with it. These chatbot robo-advisors additionally can perform other tasks such as account opening, answering queries, handling banking tasks such as transfers and bill payments,

---

<sup>10</sup> Boschke, H. (2018). Using Artificial Intelligence in Wealth Management. *The WealthTech Book*, 80-82. doi:10.1002/9781119444510.ch20

<sup>11</sup> <https://www.credenceanalytics.com/whitepapers/Chat-based-Robo-Advisors-A-Game-changer-in-the-banking-space.pdf>

### *Advantages of Chatbot over Robo advisors:*

1. Robo advisors are not personalized and moreover, they are designed to only trade on limited asset of ETFs which are predetermined.
2. The portfolio optimization process of the robo advisors is not transparent and by using chatbots we can make it completely transparent. In chatbots we can ask the investor to provide the information he wants rather than giving him some choices.
3. The interface could be made more personalized to each user over the time and we can collect other meta information regarding users that could be help for determining their persona more accurately
4. By building and developing the infrastructure for chatbots we can easily provide hybrid support of human and machine to investors. Like investors, advisors will also find it easy to interact by chatbots.

## 3.3 Platforms

There are more than 100's of platforms to choose from for creating conversational assistants from with minimum programming as possible. The right one to choose depends on each project and some of the factors to consider are an end to end solution (full cycle support), developers and community forums, analytics, languages supported, ease of creation and deploying and finally ML capabilities. There are numerous blogs<sup>12 13</sup> where some people have made comparisons. In these blogs, authors agree that IBM Watson is leading the race, and this comes to the next section where I explain why I choose IBM Watson.

---

<sup>12</sup> Retrieved February 22, 2019, from <https://medium.com/@akshay.kalkur/which-is-the-best-platform-to-develop-a-chatbot-in-seconds-d6a34bb748b>

<sup>13</sup> Retrieved February 22, 2019, [https://medium.com/@samuel\\_22971/google-dialogflow-vs-microsoft-luis-vs-ibm-watson-assistant-conversational-ai-comparision-eb2d374f1413](https://medium.com/@samuel_22971/google-dialogflow-vs-microsoft-luis-vs-ibm-watson-assistant-conversational-ai-comparision-eb2d374f1413)

## 4 IBM Watson

### 4.1 Why IBM Watson

According to the evaluation report on conversational computing platforms by the Forrester<sup>14</sup> (market research company) IBM Watson clearly leads. They evaluated seven most significant market providers on nine different criteria<sup>15</sup> which can be broadly grouped into three categories current offerings, strategy, and market presence. IBM can differentiate itself from its competitors Amazon and Google on the breadth of services provided, Application development environment, back end resources and a number of languages supported. The report also adds a note which says Amazon and Google are poised to exploit the market share by leveraging their positions in the cloud industry. IBM Watson is an umbrella term representing multiple products such as Watson assistant, discovery, studio, machine learning, tone analyzer, text to speech and speech to text.

### 4.2 IBM Watson Assistant key features

End to end platform functionality, this means IBM Watson offers everything from creating the chatbot, hosting it on a cloud and integrating it with common media platforms. During this whole process, there will not be a need for any other external services and IBM Watson provides a standalone environment that offers all tools.

IBM Watson Assistant key functionalities are (this is only a small subset of features and I included the ones I benefitted most from):

1. Documentation: Documentation is rich enough where everything is explained in a logical flow starting from simple tasks such as creating an assistant to deploying the bot.
2. Developers Community: IBM provides a platform and there are dedicated groups in stack overflow for all developers to ask common questions.

---

<sup>14</sup> Home · Retrieved February 22, 2019, from <https://go.forrester.com/>

<sup>15</sup> Koplowitz, R., & Facemire, M. (2018). *The Forrester New Wave™: Conversational Computing Platforms, Q2 2018*(Rep.).

3. **Improvement:** In Watson assistant, one can see all the previous conversations irrespective of platform deployed and the advantage is we can browse each conversation and see the intent it has been classified in to. If one notices a wrong classified intent, we can tell the assistant to retrain by providing the right intent.
4. **Flexibility:** The assistant building process can be easily coordinated among team members. One can also download the entire assistant as a JSON file which can then be re-uploaded very easily by others.

In my opinion, the IBM environment is intuitively easy to use and understand but it is still lacking on fluidity in terms of the user interface. The documentation is very overwhelming which is nicely structured from a higher level but too much information is presented at one place making it hard to find the relevant information. As such everything is available in documentation with examples provided. IBM also has a platform for learning how to build chatbots and many other new technical skills such as big data and blockchain on cognitive class <sup>16</sup>. There are many learning paths provided and I have completed the chatbot building course.

### 4.3 Limitations

The free edition which IBM named as “Lite”<sup>17</sup> has limitations in terms of functionalities such as number of API calls, number of skills or assistants and dialog nodes one can create. These are on a higher level but in detail, the lite plan doesn’t have access to intent recommendations and creating a search skill <sup>18</sup>. Despite these limitations, for an academic project, there should not be a problem unless the bot is made with the aim of deploying. If bot needs to be deployed the number of API calls and recommendations for intents will limit the performance. In addition, the customer support will not provide any kind of assistance to this Lite edition users. Finally, as claimed in some blogs the documentation cannot be easily comprehended by every individual. One should have some amount of proficiency in programming and common computer concepts such as API, JSON and other data structures.

---

<sup>16</sup> Cognitive Class. (n.d.). Retrieved February 22, 2019, from <https://cognitiveclass.ai/>

<sup>17</sup> Ibm. (2017, October 15). Retrieved April 22, 2019, from <https://www.ibm.com/cloud/watson-assistant/pricing/>

<sup>18</sup> Retrieved February 22, 2019, from <https://cloud.ibm.com/docs/services/assistant?topic=assistant-getting-started>

## 5 Methodology

### 5.1 How Chabot works

The basic building blocks of a chatbot remain the same across all the platforms but can be called with different names they are Intents, Entities and Dialog.

#### 5.1.1 Intents

Intents simply represent the purposes or goals of a user's input. In the sense of parts of speech, we can say it closely represents a verb, the action the user wants to do. For instance, some of the intents for this project include:

#Interest: when the user expresses his interest in investing and portfolio building

#Strategy: when the user expresses his interest in either following an active or passive strategy

#### 5.1.2 Entities

Entities represent a term or object in the user's input that is relevant to the intents and they add context to the intent. If intents represent verbs, entities represent nouns (object or context). Entity extraction and identification is key to modern chatbots because unless they extract entities from natural human conversations there is not much difference with earlier rule-based approaches. Entity examples:

@strategy\_overall = this entity represents the strategy either active or passive

#### 5.1.3 Dialog Flow

Dialog is a predefined story that guides the conversation with users. After classifying the intent of the response, the bot should give a response to this intent from the dialog. Dialog has all the mappings of intents and entities with responses. The sample dialog for this project is:

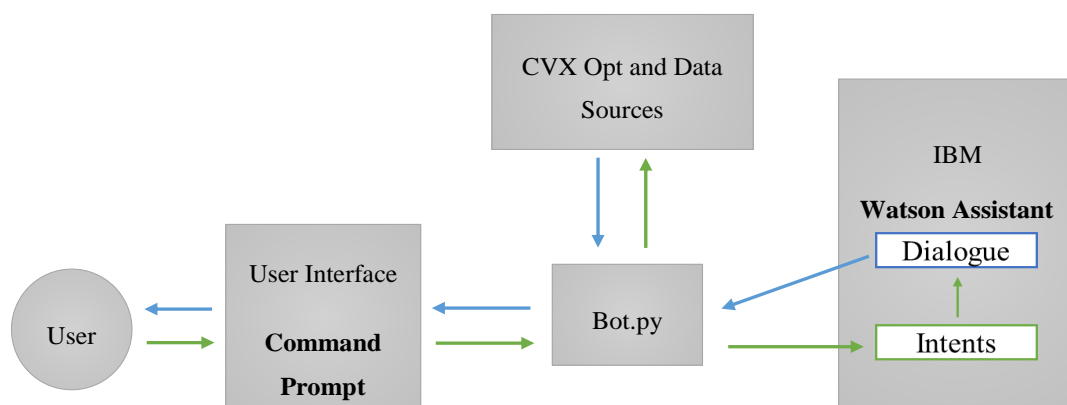
Detected intent: #General\_Greetings  
 Hi, I'm a chatbot and I can help with creating a portfolio according to your preferences. Feel free to ask me any questions on investing in a portfolio of stocks and ETF's. Please enter 'quit' to end the bot  
 >> i want to invest —————→ User Input  
 Detected intent: #interest  
 Yes, I can help you with that...!! I would like to know if you want to follow an active or passive investment strategy?  
 >> what is active strategy ? —————→ User Input  
 Detected intent: #question\_strategy  
 Detected entity: @strategy\_overall  
 Passive investment refers to the process of creating a portfolio with the sole intention of tracking a benchmark. Active investment refers to the process of creating a portfolio with the expectation to beat the market by actively making investment decisions.  
 >> I want to follow active strategy —————→ User Input  
 Detected intent: #strategy  
 Detected entity: @strategy\_overall  
 please enter the lists of stocks  
 enter the list of shares or ETF's you want to hold in your portfolio and enter 'done' in the end

**Figure 1 conversation with chatbot**

The blue boxes represent the intent the input is classified in to and the green box represents the entities identified in the input. We can see that when the user is asking the question “what is the active strategy ?” here the bot correctly classified into “question\_strategy” intent.

Intents, entities and dialog are crucial for any bot and these three are very basic concepts and there are many advanced concepts such as contextual entities, digressions and slots which are needed for building complex conversations. Details of all of these are available in the IBM documentation.

## 5.2 Flow diagram of the bot



**Figure 2 Complete Flow chart of the prototype**



The user can start interacting with command prompt as a user interface. The red arrows represent the input conversation given by the user. The conversation is sent to a python script which then transfers it to IBM Watson assistant through the API key. The most critical step is to classify the conversation to the right intent so that the right response is returned. Every intent is mapped to a specific response and this process is already developed in the dialog. The python script then receives the response and the corresponding intent it belongs to along with entities if any. If the intent is about building the efficient frontier then python script will execute the corresponding optimization framework which is developed over CVX opt library. Some entities are sent as arguments to these functions and the final computed values are returned to the command prompt to be seen by the user. The final output can be the result of the optimization or just the same response as received from the IBM Watson depending on the state of the dialogue and specific intents observed. To use the command prompt, one should simply go the folder same as the python script and type “Python bot.py”. The first message can be readily seen in the prompt and we can start the conversation.

### 5.3 Python

The whole project has been developed in python due to the availability of rich libraries and community. I have used CVX opt for all the portfolio optimization because this library syntax follows very closely to solvers in MATLAB. The important point to note is we must convert everything into cvxopt matrix objects for calling the solver and this makes the whole process intuitive for people who have experience with MATLAB. The limitation is, it has very limited support and another better open source alternative is SciPy library. For good customer support, one can use IBM CPLEX (commercial). It also has a free edition with limited features, but it has not been explored. For most of the project pandas, data frames and matplotlib visualizations are used.

## 5.4 Active Investing – Portfolio Optimization

Active investing, as the name suggests is a strategy where an investor takes a hands-on approach while creating portfolios. The investor's primary goal is to gain more returns than the market benchmarks (major market indices) and this involves deeper analysis of securities and type of the portfolio strategy to implement. Most of the present-day portfolio strategy is derived from Modern portfolio theory.

### 5.4.1 Modern Portfolio Theory

The modern-day portfolio theory is introduced by the economist Harry Markowitz as a dissertation in the year 1952<sup>19</sup>. This pioneering work by Markowitz remains the most popular way of approaching a portfolio optimization problem<sup>20</sup>. A simple search for his original paper in Google scholar shows around 38,000 citations. Markowitz framework provides an excellent way to answer the question “given an amount of money, how to allocate this from a set of possible choices”. Markowitz assumed that the return of an asset as a random variable (normal distribution) and expected the mean and standard deviation of this random variable as expected return and risk of the asset. Till the Markowitz theory, most of the previous work has not considered the risk of an asset<sup>21</sup>

Another key idea introduced by the Markowitz is the concept of diversification (do not keep all the eggs in one basket). The variance or risk of a portfolio should be evaluated based on the covariance of the assets returns. A rational investor should either try to minimize the risk for a given return or maximize the return for a given risk. The risk and return should both be considered during the portfolio selection. This risk-return trade-off can be formulated as an optimization problem. The solutions to this problem give efficient frontier and investor can choose any point in this efficient (for the given risk maximum return) frontier based on his preference of risk.

---

<sup>19</sup> Markowitz, H. (1952). Portfolio Selection\*. *The Journal of Finance*, 7(1), 77-91. doi:10.1111/j.1540-6261.1952.tb01525.x

<sup>20</sup> Zhang, Y., Li, X., & Guo, S. (2017). Portfolio selection problems with Markowitz's mean–variance framework: A review of literature. *Fuzzy Optimization and Decision Making*, 17(2), 125-158. doi:10.1007/s10700-017-9266-z

<sup>21</sup> Kolm, P. N., Tütüncü, R., & Fabozzi, F. J. (2014). 60 Years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research*, 234(2), 356-371. doi:10.1016/j.ejor.2013.10.060

### 5.4.2 Mean-Variance Portfolio Optimization

There is a number of ways one could formulate the problem for generating efficient portfolios<sup>22</sup>.

The one I adopted for this project is:

Minimize  $x^T Q x$

Subject to  $\bar{\mu}^T x \geq \varepsilon$ , decision variable is  $x$ : weights

$\mathbb{1}^T x = 1$ , represents the budget constraint

$x \geq 0$ , represents no short selling

Here ‘ $Q$ ’ is the covariance matrix and  $\bar{\mu}$  is the expected return vector and the ‘ $x$ ’ is the vector of weights, which is the fraction of total money to be invested in a particular asset.  $\varepsilon$  is the minimum expected return and for each given value of  $\varepsilon$  the above optimization problem tries to find the portfolio which has minimum variance (risk). By changing the values of  $\varepsilon$  from zero (or sufficiently large negative number) to return corresponding to a maximum return of the asset in the universe we considered (because if we invest all wealth in this asset, this is the maximum expected return for our portfolio) one can obtain different points on the efficient frontier.

The efficient frontier can also be generated by formulating the above problem as a multi-objective optimization problem where both return and risk are part of the objective function.

Minimize  $\bar{\mu}^T x + \lambda x^T Q x$

Subject to  $\mathbb{1}^T x = 1$ ,  $x \geq 0$ , decision variable  $x$ : weights

Here  $\lambda > 0$  is a risk aversion factor. Different portfolios in efficient frontier can be obtained by varying  $\lambda$  from smaller values (risk-seeking) to larger values (more risk averse).

---

<sup>22</sup> Cornuéjols, G., Tütüncü, R., & Peña, J. F. (2018). *Optimization methods in finance*. Cambridge: Cambridge University Press.

#### 5.4.2.1 Minimum Variance Portfolio

Minimize  $x^T Q x$

Subject to  $\mathbb{1}^T x = 1$ ,  $x \geq 0$ , decision variable  $x$ : weights

By removing the return constraint from general mean-variance optimization problem or by replacing ' $\epsilon$ ' with a negative number we can obtain a minimum variance portfolio. This is the portfolio that would have the lowest risk than any other weight combinations. *The point to note is, the risk of the minimum variance portfolio is always less than the risk of the asset which has the least variance amongst the asset universe we considered. This portfolio is ideal for an investor who is highly risk averse.*

#### 5.4.2.2 Maximum Return Portfolio

Maximize  $\bar{\mu}^T x$

Subject to  $\mathbb{1}^T x = 1$ ,  $x \geq 0$ , Decision variable  $x$ : weights

The answer to this optimization problem is the same as investing all your budget on the asset which as maximum return amongst the  $n$  assets. *This portfolio is ideal for investors who have no preferences for risk and only looking to maximize the return.*

#### 5.4.2.3 Maximum Sharpe Ratio

Sharpe ratios is defined as:  $\frac{R_p - R_f}{\sigma_p}$

where  $R_p$  denotes return of the portfolio,  $R_f$  risk-free rate and  $\sigma_p$  is the standard deviation of the portfolio

Maximum Sharpe ratio portfolio can be determined by

Maximize  $(\bar{\mu}^T x - r_f) / \sqrt{x^T Q x}$

Subject to  $\mathbb{1}^T x = 1$ ,  $x \geq 0$

The above objective function can be simplified by change of variables to a simpler problem<sup>22</sup>:

Maximize  $y^T Q y$

Subject to  $\mathbb{1}^T y = k$ ,  $k \geq 0$ ,  $(\mu - r_f)y^T = 1$ ,  $y \geq 0$ , decision variables vector  $y$  and scalar  $k$

Original weights can be estimated by  $x^* = \frac{y^*}{k^*}$

Sharpe ratio is a measure of the excess return of a portfolio or asset (with respect to the risk-free asset) per unit of risk. In simple terms, Sharpe ratio is the additional return you are receiving for the additional risk you hold over a risk-free asset (zero risk asset). *As Sharpe ratio is a risk-adjusted measure of return, we can use it to compare the performance of two portfolios though they have different risks. Max Sharpe ratio portfolio is ideal for investors who is looking for good tradeoff of risk with return.*

#### 5.4.2.4 Limitations of MVO

*There are many limitations to mean-variance optimization proposed by Markowitz and some of them are:*

1. Single period model and doesn't take any information of the future into account
2. Assumes normal distribution for returns which may not be the case always and also normal distribution is symmetric, and a rational investor doesn't want to minimize the variance of the portfolio in both directions (if portfolio return is higher than the average it is an advantage)
3. Highly sensitive to parameter estimation and even a small change in the estimated returns would lead to a drastic change in weight composition making the portfolio unstable
4. The portfolios determined are overly concentrated on only a few assets.

#### 5.4.2.5 Equal Risk Contribution

$$\text{Maximize } \sum_{i=1}^n \sum_{j=1}^n (x_i(Qx)_i - x_j(Qx)_j)^2$$

Subject to  $\mathbb{1}^T x = 1$ ,  $x \geq 0$ , decision variable  $x$ : weights

$x_i(Qx)_i$  measure the contribution of the risk of asset 'i' to the portfolio

Equal Risk or Risk parity portfolios are like an equally weighted portfolio (giving equal weight to all assets  $1/n$ ) except here we distribute the risk of the portfolio equally. *The obvious advantage over MVO model is we don't need to estimate expected returns and only covariance matrix (covariance matrix is 10 times less sensitive than return). Another advantage is we get more diversified portfolios in terms of risk as we are making sure all assets contribute to risk equally.* There are many simpler alternatives available to solve the above problem.

#### 5.4.2.6 Robust Optimization

For MVO problem we need to estimate the returns and there is always uncertainty in estimating returns and robust optimization deals this by adding this uncertainty in to the optimization problem itself. The major difference is the robust portfolio would satisfy all the uncertain constraints unlike the normal MVO.

We can assume two different uncertain sets for returns, ellipsoidal or box. The problem formulation below is for box uncertainty set and there is good literature available for formulating robust optimization problem

$$\text{Minimize } x^T Q x \quad \Leftrightarrow \quad \text{Minimize } x^T Q x$$

$$\text{Subject to } \mathbb{1}^T x = 1, \mu^T x - \delta^T |x| \geq R \quad \text{Subject to } \mathbb{1}^T x = 1, y \geq x, y \leq x, \mu^T x - \delta^T y \geq R$$

$\delta$  is a variable which represent the maximum distance we assume our estimated returns would be from true expected returns. It is calculated by  $\delta_i = \varepsilon * \frac{\sigma_i}{\text{sqrt}(N)}$  where  $\varepsilon$  is a sizing parameter and if we assume normal distribution for errors and we are confident that the true expected return is 95% confidence interval of estimated expected returns then we can use value of 1.96. The formulation in the right is practically used to calculate robust portfolio and auxiliary variable  $y$  is introduced to handle the absolute value term.

#### 5.4.2.7 Black Litterman

$$\bar{\mu} = [(\tau Q)^{-1} + P^T \Omega^{-1} P]^{-1} [(\tau Q)^{-1} \pi + P^T \Omega^{-1} q]$$

$$\pi = \lambda Q x_{\text{mkt}}$$

*Black Litterman model removes the problem of estimating the expected returns by using estimated market equilibrium returns and adding our views to it.* Instead of using the historical returns of the stocks expected returns are to be estimated using the above formula and then we use this returns to solve the normal MVO problem. The black litterman model basically consists of two parts, market equilibrium returns and our views. The weighted sum of these two is represented by the second term and the first term is just a normalizing constant.

P is the link matrix and q is the vector of views and  $\Omega$  is a diagonal matrix with each diagonal element representing strength of each view.  $x_{\text{mkt}}$  is the weights represented by the assets in market and  $\lambda$  is risk aversion coefficient. If we do not hold any views  $\bar{\mu} = \pi$ .

#### 5.4.2.8 CVaR Optimization

*As discussed, another drawback of MVO is that it minimizes both upward and downward movement of return.* CVaR optimization is where we only minimize the downside risk.

$$\text{Minimize } q + \frac{1}{N(1-\alpha)} * \sum_{i=1}^N y_i$$

$$\text{Subject to } l_i(x) - q - y_i \leq 0, i = 1 \dots N \quad y_i \geq 0, i = 1 \dots N, \mathbb{1}^T x = 1, x \geq 0$$

Decision variables: q, y, x (weights)

N represents the number of loss scenarios either through Monte Carlo or using historical data

$l_i(x)$  loss of the portfolio in each scenario

## 5.5 Passive Investing

In Passive investing strategy an investors main goal is to build wealth gradually and investors believe that it takes lot of effort and analysis to beat the market. So, the goal here is to create portfolios which would mimic a market index and the assumption is that markets tend to move up over the time. In this project the investor can choose the index he would like to track and the assets he wants to hold. The bot will give five different portfolios across the frontier with different tracking errors along with risk and return information.

I followed the index tracking formulation as mentioned in this paper <sup>23</sup>:

$$\text{Minimize } \frac{1}{2} * x^T Q x - \sigma_m^2 \beta^T x$$

Subject to  $\mathbb{1}^T x = 1$ ,  $x \geq 0$ ,  $\bar{\mu}^T x \geq \varepsilon$ , decision variable  $x$ : weights

Where  $\beta_i = \frac{\sigma_{im}}{\sigma_m^2}$ , ‘m’ is the market or the index

---

<sup>23</sup> Edirisinghe, N. C. (2013). Index-tracking optimal portfolio selection. *Quantitative Finance Letters*, 1(1), 16-20. doi:10.1080/21649502.2013.803789



## 6 Limitations of Chatbot

As discussed, there are many advantages to chatbots which can be broadly classified as ease, speed and convenience. But there are many factors one should consider before deploying the bot and most important one is privacy and data protection of personal and sensitive data. This is most important because investors may not be willing to share everything on a chatbot. Another important fact is chatbots may be are simple computer programs, but they demand substantial investment in IT services and infrastructure to support many needs.

Chatbots cannot answer every query and there would be situations where users/investors requests are unusual and new to the bot. For such instance's humans are more suitable and are more appropriate to help clients walk through difficult markets situations or urging them to take some actions. This hybrid approach of human and Chabot gives the best of both and imposing this ability to chatbot architecture is not hard. Chatbots generally do better on specific needs where the answers are well known. Chatbots needs to be trained periodically as bots get different queries from users over time. A chatbot is as good as the training examples it sees.

Maintaining a conversation flow is not easy and one misclassification of an intent at a later stage can make the whole dialog and data collected till that point futile. This can leave investor/customer be frustrated. There are many ways one could address these and one such is using multi-bot approach <sup>24</sup>

Finally, another important factor is to the test the bot before deploying. There are multiple factors one should consider while testing and there many platforms which offer software's to test bots.

---

<sup>24</sup> Candello, H., & Pinhanez, C. (2018). Recovering from Dialogue Failures Using Multiple Agents in Wealth Management Advice. *Human-Computer Interaction Series Studies in Conversational UX Design*, 139-157. doi:10.1007/978-3-319-95579-7\_7

## 7 Chatbot Testing

Like chatbots, its testing is not new, and the earliest known technique is Turing test<sup>25, 26</sup> proposed by Alan Turing in 1950. In Turing test, a human judge will have a conversation with some other entity which can either be a chatbot or real person. If the judge thinks the other person is human when it is a chatbot then the bot passed the test. One of the popular and first competition to implement this is Loebner Competition<sup>26</sup>

Keeping aside the debate whether the Turing test is doing more harm than good to chatbots because the participates are only trying to fool judges, companies should use other ways to do testing. Although it seems natural to pick the chatbot which tricks human in believing it is not a chatbot, the primary goal of the companies is to provide the right information to the investors or users. Providing them with the service is more important than making them to think that they are talking to a human. So instead of Turing test evaluation should be chosen based on the application and to user needs. If the chatbot is meant to provide a certain service to user, then the best evaluation is to see how may users were able to get access to the service or task<sup>25</sup>.

For financial service companies some of the factors they should test their chatbot are:

1. Conversation flow, whether the bot is following the right flow.
2. Intents matching to check if the bot can identify right intents even with small queries and spelling mistakes in them.
3. Entities recognition to make sure the bot has identified all the relevant information

The fact to note is that the chatbot testing is very different from normal software testing and there are some software's such as 'Botium' which claim to do<sup>27</sup> chatbot testing.

---

<sup>25</sup> Shawar, B. A., & Atwell, E. (2007). Different measurements metrics to evaluate a chatbot system. *Proceedings of the Workshop on Bridging the Gap Academic and Industrial Research in Dialog Technologies - NAACL-HLT 07*. doi:10.3115/1556328.1556341

<sup>26</sup> Bradesko, L., & Mladenec, D. (2012). A Survey of Chabot Systems through a Loebner Prize Competition.

<sup>27</sup> Botium. (n.d.). Chatbots are driving the industry. Retrieved April 21, 2019, from <http://www.botium.at/index.html>

## 8 Conclusions

Chatbots are going beyond traditional keyword matching to understanding context, maintain a dialog flow and this has been possible due to advancements in natural language processing, and machine learning models. Chatbots have huge potential in saving money to companies and improve their customer service. No longer are customers need to wait for days for an employee to help them. Users are not averse to using bots if they add value and companies need not focus more efforts on trying to mimic them to behave like humans.

In this project, a working prototype has been developed for personal wealth management. The bot can help investors to fetch market data and build portfolios according to their preference. The chatbots will not be a replacement to the existing financial advisors or companies but they can be a valuable tool to help facilitate their work and increase the productivity. By complementing their services with chatbots one can get best of both humans and machines.

According to me, I see the chatbots based robo advisors have more potential to grow because they bring the best of machine learning and optimization techniques to all investors big or small and now it is in hands of investors to take advantage of it. As per my knowledge I could not find any empirical study which studied the advantages of using wealth management chatbot and this could be a good potential area to explore.

One more point I would like to assert is that though there are many platforms and supporting documentation to build highly powerful conversational assistants the disadvantage is one can overlook to know how they work. This is also crucial because every method or algorithm has some limitations and it is important to understand this especially in case your chatbot is failing to complete conversations with your users.

## 9 Future Work/ Recommendations

Presently, the bot can be used to have conversations but there are many improvements that could be made before deploying for user testing. I classified all the future work I will be doing into three categories:

### 9.1 Technical Improvements: Optimization Formulation

1. All strategies are implemented without taking transaction costs into account. Transaction costs are significant and can potentially reduce the returns if taken in to account.
2. Similarly, other constraints such as cardinality and minimum holding though, trivial are not yet added and this flexibility will be provided soon.
3. Black Litterman, minimizing CVaR, Risk parity and Robust optimization models are yet to be added to the bot. These features will be more useful for experienced and active investors.
4. Index tracking has been implemented as a passive strategy but, the user must give the list of assets of his choice to track the index. There could be another index tracking problem where the user just enters the index and we choose the assets which belong to the index. I couldn't yet find open source API's which provides a list of equities that are part of an index. Once this information is obtained, we can follow the cluster-based approach as described in this paper<sup>28</sup>.
5. As already mentioned previously, through chatbots one can bring the power of the latest AI techniques to all investors irrespective of their knowledge in optimization and machine learning. In this regard, I would like to implement <sup>29</sup> index tracking using deep autoencoders. A practical application would be if an investor is interested in an index and there is no ETF corresponding to this then he can use this framework to track and invest.

---

<sup>28</sup> Dexiang Wu, Roy H. Kwon, Giorgio Costa, A constrained cluster-based approach for tracking the S&P 500 index International Journal of Production Economics, Volume 193, 2017, Pages 222-243, ISSN 0925-5273, <https://doi.org/10.1016/j.ijpe.2017.07.018>.

<sup>29</sup> Heaton, J. B., Polson, N. G., & Witte, J. H. (2016). Deep portfolio theory. *arXiv preprint arXiv:1605.07230*.

## 9.2 Functional Improvements:

1. Though I choose to go with results first approach and bypass the risk profiling, this is still very important. The problem with traditional questionnaire method of determining risk profile is that it is very subjective. One possible way to overcome this challenge would be through the power of ML and NLP. Instead of questions we can use case studies and ask investors to write their response and map this to a subjectivity calculator and then may generate questions which are more realistic to the investor. The main limitation is we need to have the training data, and this is not trivial to design and implement.
2. Leveraging IBM Watson's tone analyzer, we can monitor the tone of the investor throughout the conversations and if it reaches a critical limit or following a critical path the bot will generate a signal.
3. Providing a mechanism so that the user can upload a CSV file instead of typing all the assets. By this, we can also ask users to upload their current portfolio and determine its properties.

## 9.3 Deploying the Bot

Though I spent some time trying to deploy the bot I was unsuccessful. The main reason is my limited knowledge of web development and front end. The IBM Watson assistant can be readily deployed on a custom-made website by IBM through a single click of a button, but the main issue is all the optimization framework is developed in my python notebook. The Watson assistant is called thorough API each time a user inputs a conversation. The solution to this is to use the flask framework and deploy the bot on my localhost and I will be researching more alternatives sooner.

Once deployed we can use the Watson analytics tool to see all the conversations and retrain our bot on the misclassified ones.

## 10 Appendix

The complete code can be accessed in this link:

[https://colab.research.google.com/drive/1YmgWIjw-jbOG9DS4ChdsC\\_zlFkJ5UACM](https://colab.research.google.com/drive/1YmgWIjw-jbOG9DS4ChdsC_zlFkJ5UACM)

### 10.1 Active strategy conversation

Detected intent: #General\_Greetings

Bot: Hi, I'm a chatbot and I can help with creating a portfolio according to your preferences. Feel free to ask me any questions on investing in a portfolio of stocks and ETF's. Please enter 'quit' to end the bot

User: what is a ETF ?

Detected intent: #question\_securities

Bot: A stock is a type of security which signifies ownership in the issuing corporation. An ETF holds assets such as stocks, commodities, or bonds and generally track an index.

User: i understand, I want to invest

Detected intent: #interest

Bot: Yes, I can help you with that...!! I would like to know if you want to follow an active or passive investment strategy?

User: lets go with active

Detected intent: #strategy

Detected entity: @strategy\_overall

Bot: please list the stocks or ETF's

enter the list of shares or ETF's you want to hold in your portfolio and enter 'done' in the end

please enter 'yes' if you already know the tickers of the company else 'no'

>> yes

enter the tickers the companies and enter 'done' in the end

>> aapl

>> goog

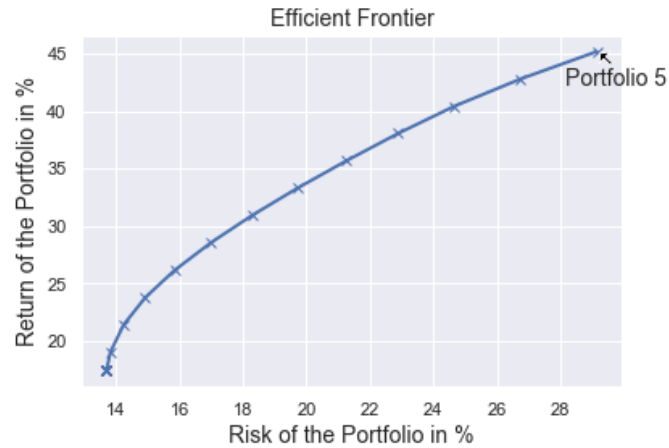
>> amzn

>> pfe

>> bx

>> dis

>> done



	1	2	3	4	5
Expected Risk in %	13	14	16	21	26
Expected Return in %	17	21	28	35	42
Sharpe Ratio	1.15	1.38	1.58	1.6	1.54

please choose the portfolio you want to move a head with further analysis, enter the number

User: four

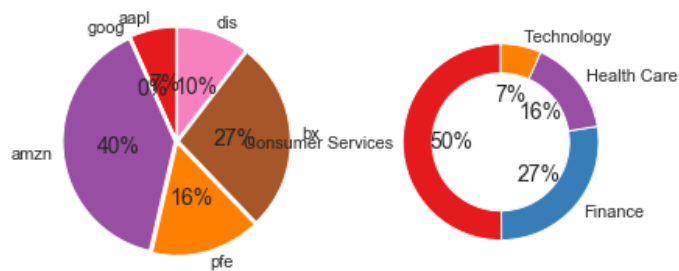
Detected intent: #portfolio\_number

Detected entity: @portfolio\_number

Back Testing of the Portfolio assuming you have 10,000 Dollars today



Weight Allocation across Assets



User: ohk thanks

Detected intent: #thank\_you

Bot: You're very welcome.

User: bye

Detected intent: #General\_Ending  
 Bot: Good bye...!!  
 User: quit

## 10.2 Passive strategy conversation

Detected intent: #General\_Greetings  
 Bot: Hi, I'm a chatbot and I can help with creating a portfolio according to your preferences. Feel free to ask me any questions on investing in a portfolio of stocks and ETF's. Please enter 'quit' to end the bot  
 User: what is a stock ?  
 Detected intent: #question\_securities  
 Bot: A stock is a type of security which signifies ownership in the issuing corporation. An ETF holds assets such as stocks, commodities, or bonds and generally track an index.  
 User: I got it, i want to invest  
 Detected intent: #interest  
 Bot: Yes, I can help you with that...!! I would like to know if you want to follow an active or passive investment strategy?  
 User: passive for now  
 Detected intent: #strategy  
 Detected entity: @strategy\_overall  
 Bot: which benchmarks would you like to follow ?  
 For example some of the famous indexes include S&P500 (ticker: GSPC) and NASDAQ100  
 enter the ticker of the index you would like to track  
 >> gspc  
 enter the list of shares or ETF's you want to hold in your portfolio and enter 'done' in the end  
 please enter 'yes' if you already know the tickers of the company else 'no'  
 >> yes  
 enter the tickers the companies and enter 'done' in the end  
 >> aapl  
 >> goog  
 >> amzn  
 >> pfe  
 >> msft  
 >> bx  
 >> dis  
 >> done

	1	2	3	4	5
Expected Risk in %	13	14	16	20	26
Expected Return in %	16	21	28	35	42
Sharpe Ratio	1.05	1.4	1.63	1.68	1.57
Tracking Error in %	7	6	9	13	20



please choose the portfolio you want to move a head with further analysis, enter the number

User: three

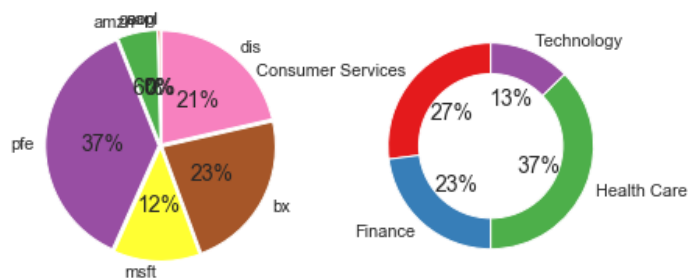
Detected intent: #portfolio\_number

Detected entity: @portfolio\_number

Back Testing of the Portfolio assuming you have 10,000 Dollars today



Weight Allocation across Assets



User: ohk thanks

Detected intent: #thank\_you

Bot: Anytime.

User: bye

Detected intent: #General\_Ending

Bot: Good bye...!!

User: quit

## 10.3 Functions

**The following function performs MVO optimization and returns the weight vector:**

```
def meanvariance(final,expected_return):
```

```
    start = datetime(2017, 1, 1)
```

```
    end = datetime(2019, 1, 1)
```

```
    plt.figure(figsize=(7, 7))
```

```
    for c in final.columns.values:
```

```
        plt.plot(final.index, final[c], lw=3, alpha=0.8,label=c)
```

```

plt.legend(loc='upper left', fontsize=12)
plt.ylabel('price in $')

returns = final.pct_change()
returns = returns.iloc[1:]
mean_returns = returns.mean()*250
cov_matrix = returns.cov()*np.sqrt(250)
n = len(mean_returns)

# Create constraint matrices
G1 = -matrix(np.eye(n))
G2 = matrix(-mean_returns).trans()
G = matrix(np.concatenate((G2, G1), axis=0))
#G = -opt.matrix(np.eye(n)) # negative n x n identity matrix
h1 = matrix(0.0, (n,1))
h2 = matrix(expected_return/100, (1,1))
h = matrix(np.concatenate((h2, h1), axis=0))
#h = opt.matrix(0.0, (n,1))
A = opt.matrix(1.0, (1, n))
b = opt.matrix(1.0)
p = matrix(0.0,(n,1))
cov_matrix = matrix(cov_matrix.values)

weights = solvers.qp(2*cov_matrix,p, G, h, A, b)
weights = array(weights['x']).tolist()
weights = [x[0] for x in weights]
return weights

```

**The following determines the efficient frontier of MVO optimization.**

```
def efficientfrontierabsolute(final,points = 20,riskfreerate = 0.0167):
```

```

    returns = final.pct_change()
    returns = returns.iloc[1:]
    mean_returns = returns.mean()*250
    cov_matrix = returns.cov()*250
    n = len(mean_returns)

    rangeofreturns = np.linspace(0,mean_returns.max(),points)
    weights = pd.DataFrame(columns = final.columns)

    G1 = -matrix(np.eye(n))
    G2 = matrix(-mean_returns).trans()
    G = matrix(np.concatenate((G2, G1), axis=0))
    #G = -opt.matrix(np.eye(n)) # negative n x n identity matrix
    h1 = matrix(0.0, (n,1))

```

```

A = opt.matrix(1.0, (1, n))
b = opt.matrix(1.0)
p = matrix(0.0,(n,1))
cov_matrix = matrix(cov_matrix.values)

for i in rangeofreturns:
#     print(i)
#     # Create constraint matrices
#     h2 = matrix(-i, (1 ,1))
#     h = matrix(np.concatenate((h2, h1), axis=0))
#     h = opt.matrix(0.0, (n ,1))
#     weight = solvers.qp(2*cov_matrix,p, G, h, A, b)
#     print(weight['x'])
#     weight = array(weight['x']).tolist()
#     weight = [x[0] for x in weight]
#     weights = weights.append(pd.Series(weight,index = final.columns),ignore_index=True)

# risk and return
returns = [blas.dot(matrix(mean_returns).trans(), matrix(x[1])) for x in weights.iterrows()]
risks = [np.sqrt(array(matrix(x[1]).trans()*cov_matrix*matrix(x[1]))[0][0]) for x in
weights.iterrows()]

All = weights.copy()
All['risk'] = risks
All['returns'] = returns
All['Sharpe Ratio'] = (All['returns'] - riskfreerate)/ All['risk']

return weights, All

```

**The following performs the MVO optimization for tracking an index**

```

def meanvariancetrackingindex(index,final):

    final = pd.concat([final, index], axis=1)

    returns = final.pct_change()
    returns = returns.iloc[1:]
    mean_returns = returns.mean()*250
    cov_matrix = returns.cov()*250
    expected_return = mean_returns[-1]
    mean_returns = mean_returns[:-1]

    beta = cov_matrix.iloc[-1,:][:-1]/cov_matrix.iloc[-1,:][:-1]
    cov_index = cov_matrix.iloc[-1,-1]
    cov_matrix = cov_matrix.iloc[:-1,:-1]

```

```

n = len(mean_returns)

# Create constraint matrices
G1 = -matrix(np.eye(n))
G2 = matrix(-mean_returns).trans()
G = matrix(np.concatenate((G2, G1), axis=0))
#G = -opt.matrix(np.eye(n)) # negative n x n identity matrix
h1 = matrix(0.0, (n,1))
h2 = matrix(expected_return/100, (1,1))
h = matrix(np.concatenate((h2, h1), axis=0))
#h = opt.matrix(0.0, (n,1))
A = opt.matrix(1.0, (1, n))
b = opt.matrix(1.0)
p = matrix(beta*cov_index)
cov_matrix = matrix(cov_matrix.values)

weights = solvers.qp(2*cov_matrix,p, G, h, A, b)
weights = array(weights['x']).tolist()
weights = [x[0] for x in weights]
return weights

```

**The following function determines the efficient frontier with respect to tracking an index by minimizing tracking error**

```
def efficientfrontierrelative(final,index,points = 20,riskfreerate = 0.0167):
```

```

    finaltemp = pd.concat([final, index], axis=1)

    returns = finaltemp.pct_change()
    returns = returns.iloc[1:]
    mean_returns = returns.mean()*250
    cov_matrix = returns.cov()*250
    expected_return = mean_returns[-1]
    mean_returns = mean_returns[:-1]

    beta = cov_matrix.iloc[-1,:][:-1]/cov_matrix.iloc[-1,:][:-1]
    cov_index = cov_matrix.iloc[-1,-1]
    cov_matrix = cov_matrix.iloc[:-1,:,-1]

    n = len(mean_returns)

    rangeofreturns = np.linspace(0,mean_returns.max(),points)
    weights = pd.DataFrame(columns = final.columns)

```

```

G1 = -matrix(np.eye(n))
G2 = matrix(-mean_returns).trans()
G = matrix(np.concatenate((G2, G1), axis=0))
#G = -opt.matrix(np.eye(n)) # negative n x n identity matrix
h1 = matrix(0.0, (n, 1))
A = opt.matrix(1.0, (1, n))
b = opt.matrix(1.0)
p = matrix(beta*cov_index)
cov_matrix = matrix(cov_matrix.values)

for i in rangeofreturns:
#     print(i)
    # Create constraint matrices
    h2 = matrix(-i, (1, 1))
    h = matrix(np.concatenate((h2, h1), axis=0))
    #h = opt.matrix(0.0, (n, 1))
    weight = solvers.qp(2*cov_matrix, p, G, h, A, b)
#     print(weight['x'])
    weight = array(weight['x']).tolist()
    weight = [x[0] for x in weight]
    weights = weights.append(pd.Series(weight, index = final.columns), ignore_index=True)

    # risk and return
    returns = [blas.dot(matrix(mean_returns).trans(), matrix(x[1])) for x in weights.iterrows()]
    risks = [np.sqrt(array(matrix(x[1]).trans()*cov_matrix*matrix(x[1]))[0][0]) for x in
weights.iterrows()]
    trackingerror = [np.sqrt(array(matrix(x[1]).trans()*cov_matrix*matrix(x[1]))[0][0] -
matrix(beta*cov_index).trans()*matrix(x[1]))[0][0] for x in weights.iterrows()]

    All = weights.copy()
    All['risk'] = risks
    All['returns'] = returns
    All['Tracking Error'] = trackingerror
    All['Sharpe Ratio'] = (All['returns'] - riskfreerate)/ All['risk']

return weights, All

```