Universitat Pompeu Fabra Barcelona

Universitat Pompeu Fabra

Enginyeria Informàtica

Automation of risk profiling and asset allocation processes with machine learning

---

Treball de Fi de Grau

Author: Simone Gigante Gassó
Director: Jordi Taixés

June 2022

Preface

This end-of-degree project has consisted of the execution of different clustering algorithms in order to be able to classify the population based on their risk profiles in order to later, through Reinforcement Learning models, be able to execute optimal investment strategies for the future. risk tolerance level of each individual.

Este trabajo de fin de grado ha consistido en la ejecución de distintos algoritmos de clustering con el fin de poder clasificar a la población en base a sus perfiles de riesgo para posteriormente, mediante modelos de Reinforcement Learning, poder ejecutar estrategias de inversión optimas de cara al nivel de tolerancia al riesgo de cada individuo.

Aquest treball de fi de grau ha consistit en l'execució de diferents algorismes de clustering per tal de poder classificar la població en base als seus perfils de risc per posteriorment, mitjançant models de Reinforcement Learning, poder executar estratègies d'inversió òptimes de cara al nivell de tolerància al risc de cada individu.

Contents

List of Figures

1 Introduction

Asset allocation is one of the essential components of wealth management activities. It consists of advice to a client on the best investing strategy considering his risk profile based on his actual position in personal and economic terms as well as their goals for the future and the general behavior of the markets. It is a complicated task because there are many factors to take into account, and in fact there is no standard in the industry to make this type of prediction despite the insistence of the institutions to regulate and standardize this kind of process to make it more transparent for consumers. In order to predict the risk profile, the clients of the advisors usually fill in forms with standardized questions and that sometimes do not reflect the real situation of the client since they do not take into account relevant information. The advisor then takes this information and develops an investment strategy that on several occasions does not correlate with the ideal level of risk previously predicted. The objective of this work is to try to simplify the full process automating it with the help of machine learning.

With this goal in mind, this task has been divided into two phases. The first phase is focused on identifying the customer's risk tolerance. For this, demographic, financial and psychological traits extracted from the database containing the results of the 2017 Financial Survey of Spanish Families have been considered. During the process, different techniques have been used to reduce the dimensionality of the data as well as for its visualization. First the data has been duly processed selecting those features that are most useful according to the current financial literature, and combining some of these features so that they have even more relevance, while simplifying the task. Some outliers in the financial features have been eliminated to improve the clustering process. For the visualization of the data, classic Python plots have been used, as well as the algorithm for the visualization of data of more than three dimensions t-SNE.

The next step has been to evaluate different clustering algorithms trying to tune the parameters of each one as best as possible in order to choose the best one. Finally, spectral clustering has turned out to be the most efficient procedure for the research data. Then the clusters have been analyzed and the main characteristics of each one have been extracted. Comparing these characteristics with the current literature on risk tolerance, each group has been provided with a certain level of tolerance to market volatility. Among the different clusters, five different risk profiles have been detected. To further personalize the risk tolerance level of each individual, the z-score of each feature within each cluster has been calculated to assess the total deviation between the responses of an individual and the mean of the values of its cluster. This z-score has been added to the cluster score to create a final and individual score.

The next phase has consisted in the application of reinforcement learning techniques to create an agent capable of organizing a portfolio containing the risk below the limit assigned to each individual in the previous clustering phase. To simplify the task, only two types of assets have been considered for the asset distribution model within the portfolio. These assets are: stocks and bonds. A personalized environment has been created in which an agent can change the distribution of assets within the portfolio and is rewarded or punished based on the portfolio's return and the risk assumed in relation to a certain risk tolerance level. The agent can see the prices of the instruments in a given number of days and the value of the portfolio at any given time, and based on this, takes actions in the form of distribution weights. The result is a distribution of the weights of each asset in the portfolio.

A small graphical interface has been developed in the form of a web application so that a user is able to carry out the entire process from the internet. From this interface it is therefore possible to fill in the necessary information to identify the level of risk of a user, and later see the best distribution of assets taking into account the maximum level of risk tolerated by the user.

In conclusion, the objective of this final degree project was to verify that thanks to machine learning techniques It was possible to automate these complex and sometimes opaque processes such as risk profiling and asset allocation and has achieved its goal. With these methods, and the graphical interface provided, a user could obtain a realistic estimate of her risk profile and an investment strategy consistent with that risk profile, without the intervention of any human advisor. Despite this the parameters used in the different processes, as well as the criteria for selecting certain elements, could improve with greater knowledge in the financial sector. Therefore this does not mean that human intervention is no longer necessary in this type of process, but it is a good starting point for an initial orientation of the user.

2

Risk profiling

Preparing an individual's risk profile is the first step in the financial advice process, since the suitability of a financial product is usually more related to the characteristics of the client who is going to purchase the product than to the product itself. [(Davies and Brooks 2014)] The elaboration of this type of profiles is complicated since they are a combination of objective and subjective characteristics of a person. Experts disagree on the number of elements that make up the risk profile, but agree that risk aversion and risk capacity are two fundamental elements for analyzing an individual's risk profile. (Grable, 2017[9], Klement, 2015[15]). Risk aversion is the investor's willingness to take financial risk and the degree of psychological pain the investor experiences when faced with a financial loss. Its inverse is risk tolerance, and it is the subjective limit that an individual is willing to tolerate without suffering the aforementioned emotional pain. The risk capacity, on the other hand, is the objective capacity of an individual to assume a financial risk based on his economic circumstances. The usual method for developing risk profiles is through the use of questionnaires.

2.1 Data preprocessing

The data used to classify people based on their risk profile has been extracted from the Family Financial Survey (Encuesta Financiera de las Familias, EFF), an official survey carried out by the Bank of Spain every three years to analyze the financial situation of Spanish households. The data used is the one from the last available survey, carried out in 2017. The data was originally separated into two different databases. The first database contains the responses of 6,413 Spanish households (rows) to 1,283 questions (columns). In this section the questions are about both demographic and economic aspects of the household. The second database contains only the answers to section 6 of the questionnaire, which only addresses questions related to the income and occupation of the respondents. This second table contains 6,413 rows corresponding to the number of households and 1,586 columns corresponding to the different questions in section 6. The first step in the preprocessing of the data has been to choose the questions that can best help us identify a person's risk profile. This reduces the dimensionality of the problem, facilitating the clustering process, and optimizes the final result by choosing truly significant features for our specific problem. Different literature has been used to optimize this part. Education is usually positively related to an individual's risk tolerance, as their risk profile becomes more conservative with increasing age (Sung and Hanna, 1997)[23]. Single investors tend to be more risk tolerant, and greater economic availability favors more aggressive risk profiles(Hallahan et al., 2004 [13]). The gender has not been taken into consideration since the most current literature doubts that it is relevant to the risk profile, arguing that previous works that defend the opposite usually have old data from another time in which women did not participate. so actively in the field of finance. Initially, 113 categories from the first database have been selected and then they have been combined with each other to make them even more relevant while reducing the dimensions of our problem. In the case of the second table it seems that it has a large number of null values and therefore may be of little use to us, however this happens because many ways of obtaining profits are contemplated. For this reason I have decided to put it all together in a single variable, thus achieving a category that groups all the monthly income of a person. This category has non-null value for all rows after that. After having gathered all the questions by category (loans, investments, properties, income, expenses and pension plans) the final result has been the fields shown in Table 1. It is necessary to take into account that some questions are addressed to the main member of the household that is answering the survey and others refer to all the members of the household. This somehow reduces the value of our data since we do not know if, for example, who is responding to the survey is who is in charge of the family economy. This situation can lead to the existence of some outliers. To deal with the outliers, the Z-score of each of the financial fields and has been calculated and then every row containing values that exceed three times the standard deviation have been eliminated.. Finally, the linear correlation between variables has been analyzed. Correlations go from -1 to 1, indicating a positive or negative correlation with the predicted variable. In this case we can see that there are not big linear correlations ($0.5 <$ or > 0.5). That doesn't mean that the data is not related, but just that their connection is maybe more complex.

Field	Values
Age	$19 <= x <= 85$
Marital status	$1 <= x <= 2$
Maximum education level achieved	$1 <= x <= 8$
Value of the main residence	$0 <= x <= 1.3 * 10^6$
Future expenses	$1 <= x <= 3$
Percentage you would spend in the next 12 months if you won a lottery	$0 <= x <= 100$
Financial risk you are willing to take when you save or make an investment	$1 <= x <= 4$
Total amount allocated each year to pension plans	$0 <= x <= 5.100$
Total value of pension plans	$0 <= x <= 147 * 10^3$
Total monthly income	$0 <= x <= 10.400$
Total monthly household expenses	$-7.264 <= x <= -64$
Total amount of money in bank accounts	$0 <= x <= 800 * 10^3$
Total amount owed on loans still to be paid	$-530 * 10^3 <= x <= 0$
Total value of investments	$0 <= x <= 16,4 * 10^6$
Annual investment income	$0 <= x <= 638 * 10^3$

2.2 Clustering

Clustering is the task of identifying similar instances and assigning them to clusters, or groups of similar instances (Géron, 2019 [11]). The difference with traditional classification is that clustering is an unsupervised task because the data is not labeled. There are different clustering algorithms that approach the problem differently, and the best option depends of how the data is structured. As in this case the data is in 16 dimensions, the global structure is difficult to visualize, although some strategies applied for visualization will be shown later. However, at this point I have chosen to test some of the most popular algorithms in the different clustering methodologies with the intention of finding the one that best solves the task of this work. In this case, the problem has been approached as a customer segmentation task in order to divide individuals into different groups and once the groups have been obtained, analyze them in detail with the current literature on risk profiles, risk tolerance and risk capacity in order to be able to classify the groups based on their risk profile (from more conservative to more aggressive). To carry out the clustering, one more step has been done in the data preprocessing, scaling the features using the MinMax function present in the sklearn library. Machine learning algorithms do not usually perform well when the features have numerical values at very different scales, as in this case in which economic attributes such as the value of houses or investment portfolios are much higher than the values of other categories such as age or education. The MinMax function compresses the values on a scale from 0 to 1, and does so by subtracting each item from the minimum value in that category, and dividing the result by the subtraction between the minimum and maximum value in that category.

2.3 Evaluation metrics

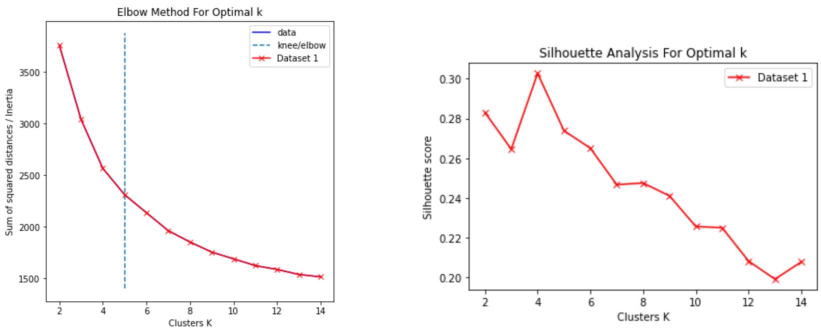
There are different metrics to evaluate the quality of the clustering work performed by an algorithm. Throughout this work, several have been used for the optimization of the different algorithms. Specifically, the Silhouette score [20], Calinski Harabasz index [4], Davies Bouldin index [8] and S_Dbw index [12] have been taken into account. However, as showed by Liu et al. (2013) [18] the best metric to compare different clustering algorithms is S_Dbw. This happens because S_Dbw is the only validation metric that considers all the five different relevant aspects during clustering: monotonicity, noise, density, subclusters and skewed distributions. S_Dbw measures the inter-cluster separation, that is the calculation of the average density in the region among clusters in relation with the density of the clusters, and intra-cluster variance, that is to calculate how spread out the clusters are on average in terms of variance. The index is the addition of these two terms and the minimum value of S_Dbw indicates the optimal cluster number.

2.4 Data visualization with t-SNE

In order to visualize the data, the t-SNE (t-distributed stochastic neighbor embedding), Van der Maaten and Hinton, 2008 [25] , algorithm has been used. This algorithm allow interpreting n-dimensional data for $n >= 3$ in two-dimensional graphs. This has been useful to be able to visualize the quality of the clusters created even if it has to be considered that some information is lost when reducing the number of dimensions. The t-SNE algorithm calculates for each data point the distance between them, and projects that distance on a normal curve centered on the point in question. That distance between the points and the curve is called unscaled similarity. Once all of these unscaled similarities have been calculated, they are normalized so that their sum is equal to one. Subsequently, the points are randomly projected to a lower dimension and the previous process is repeated, but this time the unscaled similarities are calculated on a t-distribution. Once this step has been completed, the algorithm has two distance matrices and will move the points little by little so that the second matrix (obtained on the t-distribution) resembles the first original matrix as much as possible. In the Sklearn documentation it is recommended to use some other dimensionality reduction algorithm such as PCA if the number of features is very large ($n > 50$). This is not the case, so t-SNE has been applied directly to the data.

2.5 K-means

K-means is one of the most popular algorithms for clustering. It is a centroid based method which means that a centroid is created for each cluster and each data point is assigned to the cluster whose centroid is closest. Once all the points have been assigned, the position of the centroids is updated so that they are once again in the center of their group and the data points are re-assigned to each group depending on the new distances. This happens iteratively until the algorithm converges and the centroids no longer move.



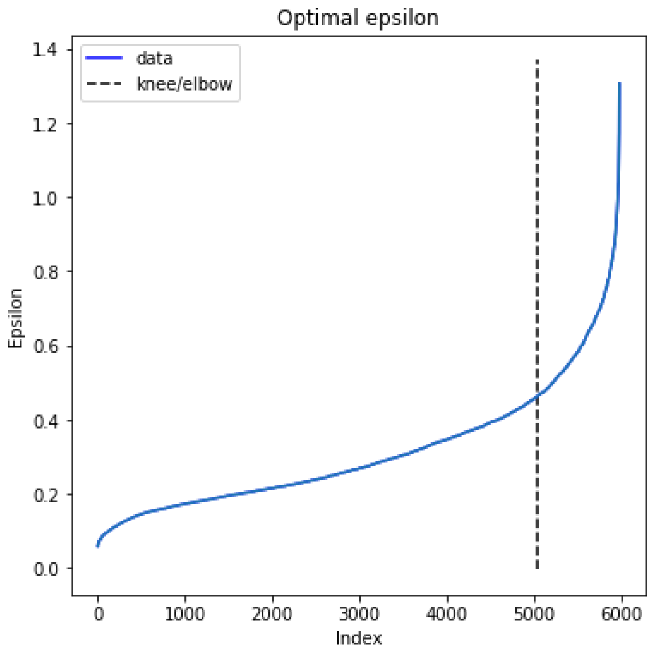
(a) (b)

Figure 2.1: Finding K number of clusters for K-means (a) Elbow method (b) Silhouette score

The main problem with this algorithm is that the number of clusters has to be specified at the beginning, and in this case the total number of clusters is unknown. Two different techniques have been used to find the optimal number of k. In the first place, it has been applied what is usually called the "elbow technique" and consists of looking for the number of clusters in which the inertia curve flattens. The inertia is calculated by measuring the distance between each data point and its centroid, squaring this distance and then adding these squares across one cluster. The second method used for K-means optimization is measuring the silhouette score for different clusters. The silhouette score is the mean silhouette coefficient over all the instances and to calculate the silhouette coefficient we do the following. First we compute the difference of the mean distance to the instances of the next closest cluster and the mean distance to the other instances in the same cluster. Then the result of this difference is divided by the maximum value between the mean distance to the instances of the next closest cluster and the mean distance to the other instances in the same cluster [11]. As Figure 1 shows, the optimal number of clusters for K-means in our data is between 4 and 5 clusters. Both values have been then tested.

2.6 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density based algorithm which means that it looks for densely populated areas of data points to assign clusters. DBSCAN uses the *coreinstance* idea to find the clusters. Core instances are data points that have at least n data points in a radius less than or equal to a distance ϵ. All the points that are inside the circle of radius ϵ and center a core instance, form a $\epsilon - neighborhood$. Once all the core instances have been found, the first cluster is assigned to one of these core instances and all the core instances that are within a distance of ϵ are added to that cluster. This process is repeated with new core instances and so on iteratively until no more core instances are added. A new cluster is then assigned to a core instance that does not yet have a cluster and the above process is repeated. When all the core instances belong to a cluster, the points that are not core instances but that are at a distance less than or equal to ϵ are added to each cluster. However, since these points are not core instances, they are not used to search for new points. Once this step is finished, the remaining points that have not yet been assigned to any cluster are considered anomalies. The two main problems encountered in this case are that the minimum number of neighbour points n and the maximum distance ϵ have to be defined by the user, and this is not easy to see with so many dimensions; the second problem is that the algorithm classifies some users as anomalies and puts them in the same group, but in the case of risk profiling this may not make sense since someone with an extremely high financial situation could be in the same group as a person in a situation of extreme poverty, when obviously their financial risk profiles are very different. Schubert et al. (2017) [22] suggest that the parameter n should be the double of the number of dimensions of the data for large dimensional data, but as a rule of thumb for datasets with less than 30 dimensions, n can be the same as the number of dimensions of the data. In this case I have tried both solutions and the one that better worked is $n = 15$ that is the number of dimensions of the dataset. To find the parameter ϵ, Rahmah and Sitanggang (2016)[19] Schubert et al. (2017)[22] and Sander et al. (1998)[21] suggest that it cat be found by calculating the distance to the nearest n points for each point, sorting and plotting the results and then identifying the point where the change is most accentuated (similar to the elbow method). I have used the NearestNeighbors and KNeighbours methods for this task with again



(a)

Figure 2.2: Finding ϵ distance value for DBSCAN with NearestNeighbors method.

$n = 15$ that is the number of dimensions of the dataset. As the Figure 2 shows, the optimal value of ϵ is 0.466. To solve the problem of anomalies, I have removed them from the data once they have already been labeled by the DBSCAN algorithm, and applied a KNeighbors classifier to the data without the anomalies, passing the anomalies as new points. By doing this each anomaly has been assigned to the nearest cluster and all points are within some cluster.

2.7 OPTICS

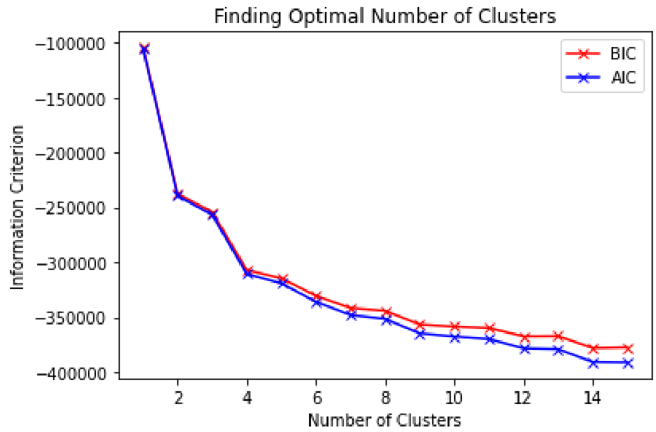
OPTICS (Ordering Points To Identify the Clustering Structure) was introduced by Ankerst et al. [2] and it can be considered as an improved version of DBSCAN algorithm. The approach its similar to the one of DBSCAN but it capable to detect clusters in data of varying density.To achieve this, introduce two new parameters: *Core distance* and *Reachability distance*. The core distance of a p point is the smallest distance ϵ to a point in its $\epsilon - neighborhood$. If p is not a core point, then it's core distance is undefined. The reachability distance between a point p and another point is the maximum value between the distance of this two points or the core distance of p. With the OPTICS algorithm from the Sklearn library, we only need to specify n to find the reachability distance and the core distance. As the Sklearn documentation specifies, their implementation deviates from the original OPTICS by first performing k-nearest-neighborhood searches on all points to identify core sizes, then computing only the distances to unprocessed points when constructing the cluster order.[1] Clusters have been then extracted using the automatic xi method proposed in the original paper.[2] In this case we also have to specify both n and ϵ. The same values used in DBSCAN have been used and the result has been very similar.

2.8 BIRCH

The BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) algorithm introduced by Zhang et al. (1996) [28] was first designed specifically for very large datasets [11]. BIRCH converts data into a tree data structure using the tree leaves as centroids. This final tree can be then the input for another clustering algorithm. The main parameter that has to be specified is the number of clusters. The number of clusters for which the index S_Dbw gives the minimum value has been chosen, in this case $n = 8$ clusters.

2.9 Gaussian mixtures

As its name indicates, clustering by Gaussian mixtures is applied assuming that the data are the result of different Gaussian distributions. These types of clustering models require you to specify the number of clusters present in the dataset. To find the number of clusters k Geron (2019) [11] suggests two methods. The first one is by finding the k value that minimizes the value of a theoretical information criterion like the Akaike information criterion (AIC) or the Bayesian information criterion (BIC). These models are commonly used in statistics for model selection, and the main criterion they use is the amount of information lost by each model. In this case I



(a)

Figure 2.3: Finding k number of clusters for Gaussian mixture model with AIC and BIC.

have calculated the value of both metrics for different values of k and the result is $k = 14$ as it can be seen in figure 3. The other technique is through a Bayesian Gaussian Mixture that, given n greater than the number of expected clusters, distributes the probabilities that a data point falls into one cluster or another. Once you have the weights, you can dispense with those clusters that do not reach a weight of 1%.

2.10 Agglomerative clustering

The AgglomerativeClustering class from Sklearn performs a hierarchical clustering using a bottom up approach: each observation starts in its own cluster, and clusters are successively merged together. There are different kinds of linkage criteria that determines the merge strategy. In this case, the use of dendrograms has been used for each different linkage method. For this step, the methods of the scipy library dedicated to hierarchical clustering have been used. The dendrogram illustrates how each cluster is composed and provides a color map to assess the number of clusters. The different results can be seen in Figure 4. Analyzing the number Although the dendrogram specifies a specific number of clusters, we can analyze how many elements are in each cluster and cut those that have very few elements.

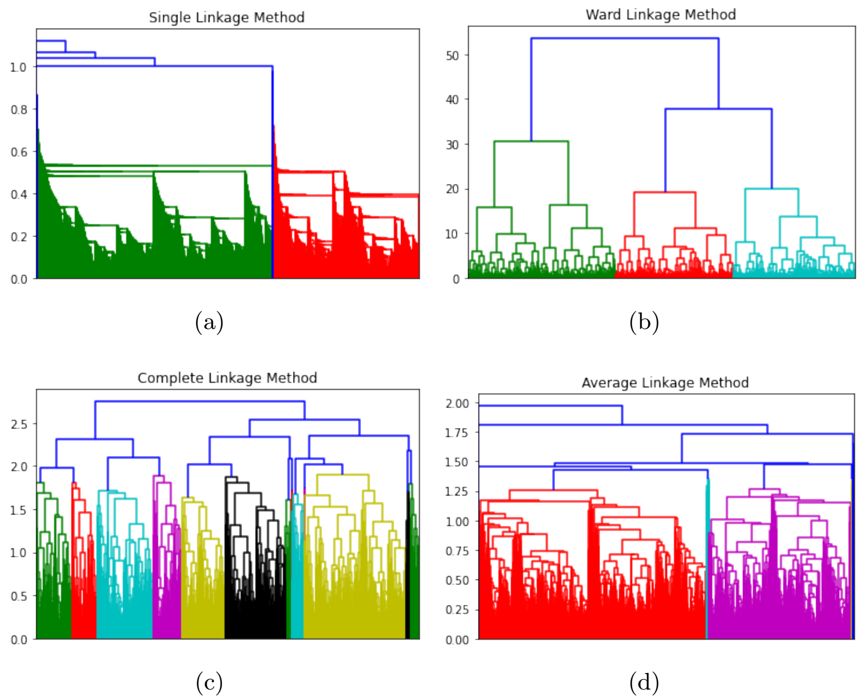


Figure 2.4: Dendrograms for different linkage methods (a) Single (b) Ward (c) Complete (d) Average

In the case of using *average* linkage for example, the algorithm returns 7 clusters, but of those seven, two clusters have > 2000 elements each, while the other five have < 20 elements each. In the end, 3 clusters have been considered for ward linkage, 6 clusters for complete linkage, 2 clusters for single linkage and 3 clusters for average linkage.

2.11 Spectral clustering

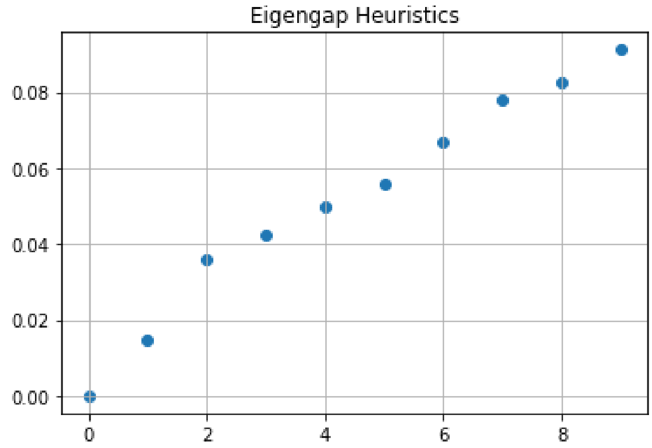
Spectral clustering is capable to capture more complex structures than other clustering algorithms and that is why today it is one of the most popular clustering methods. It needs to be given the expected number of clusters k. It uses the eigenvalues of the adjacency matrix to reduce the dimensionality of the data and then it applies the clustering with some cluster algorithm. In this case, the Sklearn library algorithm has been used, which uses K-means to cluster the data once they have been reduced. The adjacency matrix can be created using k-nearest neighbors or using a Gaussian function (RBF). In this case, a plot of the value of S_Dbw has been made for both and it is always slightly lower in the case of k-nearest neighbors, so this has been the chosen option. This algorithm needs the number of clusters k in advance. To find it, it has been took into consideration the work of Zelnik-Manor and Perona (2004) [27] and Von Luxburg (2007) [26]. In their paper, their propose an analysis of the eigenvalues of the Laplacian matrix to find the optimal value of k. Zelnik-Manor and Perona show how it is possible to calculate the adjacency matrix from the n nearest neighbors and Von Luxburg discovered a technique called Eigengap Heuristic to find the optimal number of k. This technique consists first of all in calculating the eigenvalues of the normalized Laplacian matrix L obtained thanks to the adjacency matrix of the previous step. Once the eigenvalues are available, they are ordered in increasing order, and for each eigenvalue e the distance with the next eigenvalue $e + 1$ is calculated. This distance is in absolute value. Once all the distances are found, the largest gap between the eigenvalues is sought. In this case, the number of eigenvalues has been limited to the first 10 so that the clustering is accessible for interpretation. As can be seen in Figure 5, the largest gaps can be found in eigenvalues 6 and 7. These will be the different values of k that will be tested.

2.12 Mean-Shift

During the realization of this research, experiments with the Mean-Shift algorithm [7] have been carried out, but the results have been so poor that they do not deserve to be analyzed.

2.13 Results

Once the clusterings have been obtained, the data of each one of them has been analyzed to extract the different profiles. For each risk level, a maximum turbulence percentage has been assigned, which will later be affected by the z-score of each member evaluated within its own cluster. The results are as follows, they can be seen summarized in the table at the end of the section.



(a)

Figure 2.5: Plot of the first 10 eigenvalues of L for the Eigengap Heuristic method

Cluster 0: MEDIUM RISK TOLERANCE. The majority of the members of cluster 0 are adults in their working age, and around the middle of their vital cycle, all of them married and divided almost at the half between people with mandatory education as their maximum titulation and people with non obligatory studies. More than the 40% of the people in this cluster has a lower income than the minimum spanish salary that is (1100€) and almost 3/4 parts has an income equal or below the 2000€. They have a high amount of savings in general. The debt proportion is almost 50/50 but more than the 70% has small debts and just the 9% has big debts. Their familiar expenses are higher than their incomes what it probably means that there are other salaries at home. A medium value of the members have retirement plans, and more than the 80% are house owners of houses with medium or high value. Almost the 100% thinks that their future expenses will increase. The 30% of them has investing experience with medium-high amounts. From a behavioral point of view they have high or medium high risk aversion and they are average thrifty.

Cluster 1: MEDIUM-HIGH RISK TOLERANCE. This group it's the second youngest and around the 80% of their members are in their working age with 35% younger than 50. All of them are single. The 80% of them has non mandatory degrees as they maximum study level, the second biggest group is the 11.6% of members with university studies, is the group with more studies. Is the second group with highest salaries and the third in savings what it it may mean that they are a bit more spendthrift than average or that they invest more money. The 60% is debt free and the 75% is debt free or has small debts. Their expenses are in line with their income or are even lower than they could be, they are also the second group in terms of experience in investments as well as in amounts invested (first in median) and benefits per investment, so the theory that members of this group allocate a good part of their income to their investments to increase their assets seems to be confirmed. Big majority has no debt or small debts and is the second group in terms of retirement plans savings. The 75% has a house, in general with a medium-high or high value and their expenses in the future are expected to decrease in the majority of the cases, or stay the same in the rest. From a behavioural point of view the 22% has a medium-high risk aversion and the 73% high risk aversion.

Cluster 2: MEDIUM-LOW RISK TOLERANCE. It's the youngest group with a 38% of the people younger or equal than 50 years old (5% <= 30), and the big majority is in their working age. All of them are single. Almost the half has mandatory studies as their maximum, and another 46.3% has pre-university studies as their maximum. Their salaries are low in the half of the members and medium-low in another 30%. That means than more than the 80% has low or medium low salaries. The 33% has almost no savings and another 21% has low or medium low savings in their bank accounts. The 82% is debt free or has a low debt amount and for the 16% that has retirement plans they savings amount in this field is medium. Their behavioral values are average. Just the 20% has investments with an average return of the 3%. The 70% has a house with a low or medium value in the majority of the cases. Their income will increase in the future.

Cluster 3: HIGH RISK TOLERANCE. The people in this cluster is in the majority of the cases in the last part of their work cycle, all of them married and the big majority has non obligatory studies, with the second biggest amount of university degrees representing the 27% of all the university degrees. Members of cluster 3 have the highest salaries, with a big amount of savings in their bank accounts, more than the 30% has pension plans, almost the half has already investing experience and many of them receives an income from those investments. Their familiar expenses are higher than their incomes what it probably means that there are other salaries at home. They have the biggest debt values, but almost the 60% is debt free and another 13% has small debts. From a behavioral point of view is the group with less risk aversion with around a 30% of people with medium risk aversion.

Cluster 4: LOW RISK TOLERANCE This is an old cluster with the majority of the people over 70 years old, all of them single and all of them with mandatory education as their highest studies level. The 70% has a low income and the 95% a low or medium low income. Half of the members of this cluster have almost no savings and another 20% has medium-low salaries. The 80% is debt free and another 12% has very low debts. Almost no one has very big debts (0.6%). Their expenses are low, in line with their income, and even somewhat lower so they could be the only income in their household, which can be supported by the fact that they are all single. Almost no one has retirement plans and no investing experience. The 80% that has a house has a low or medium-low value house, and their expenses in the future will decrease. From a behavioral point of view they are very risk averse and in the case of winning the lottery they would spend +90% in the first year which may mean that they do not handle money well.

Cluster 5: MEDIUM-LOW RISK TOLERANCE. They have same studies levels than cluster 4 and the big majority of the people is between 50 and 70 years old what it means that they are in the middle or last part of their working cycle. They are similar to cluster 4 in senses of income and savings even if cluster 5 values are a bit higher, but they have also bigger debts and expenses what may mean that there are more incomes in their households, also because they are all married. More people has retirement plans in comparison to cluster 4 and slightly more valuable houses as well as they have more investing experience.

Cluster 6: LOW RISK TOLERANCE. Cluster 6 is very similar to cluster 4.

Cluster	**Risk**
0	Medium
1	Medium - High
2	Medium - Low
3	High
4	Low
5	Medium - Low
6	Low

Figure 2.6: t-SNE representations of the different clustering results (a) K-means for $k = 4$ (b) K-means for $k = 5$ (c) DBSCAN (d) OPTICS (e) OPTICS after near neighbors (f) BIRCH (g) Gaussian Mixtures with n components $= 14$ (h) Gaussian Mixtures with n components $= 12$ (i) Bayesian Gaussian Mixtures (j) Agglomerative Clustering with complete linkage and $n = 9$ clusters (k) Agglomerative Clustering with complete linkage and $n = 10$ clusters (l) Agglomerative Clustering with complete linkage and $n = 11$ clusters (m) Spectral clustering with $n = 6$ clusters (n) Spectral clustering with $n = 7$ clusters

3 Asset allocation

One of the characteristic processes in the wealth management area is asset allocation. Once a person's risk profile is known, an investment strategy is drawn up, and the amount invested in the different types of financial instruments is calibrated based on the risk level of the financial instruments and the client's risk profile. There are three different types of financial instruments: securities, derivatives and commodities. Securities are a type of instrument that is used to directly finance companies, banks, public entities, or governments. Essentially, securities represent an entitlement to something, like an asset or a contract. This category includes stocks, bonds and mutual funds. This research has focused on securities, but its main concept could be extended to derivatives (like futures and swaps) or commodities such as gold.

3.1 Reinforcement Learning and finance

Reinforcement Learning is a branch of machine learning in which the model learn how to act in a certain environment by trial and error, learning the strategies that result in a higher reward (Sutton and Barto 2018) [24]. The field of Reinforcement Learning is constantly expanding, as well as the number of research works that try to apply it to problems in the financial field. Different literature has helped and been an inspiration for this project. In the first place, all of this second part has been carried out using different tools present in the FinRL library introduced by Liu et al. (2020) [17]. Different papers have used the power of Reinforcement Learning for portfolio optimization: Jiang et al. (2017) [14] Zhang et al. (2020) [29] Guan et al. (2022) [10]. However, all of them try to create a model that provides a single strategy that is the most successful from a returns point of view. Instead, the objective of my project is to design a model that takes into account a person's risk profile and provides, based on that, a personalized strategy. The motivation behind this idea is that the most important thing in an investment is that the investor is comfortable with it. Obviously, the final returns are important in an investment, but so is the volatility associated with that investment. In other words, even if a strategy is very good in the long term, if at any point the investment falls more than the investor can bear, the investor is likely to withdraw and lose money. The objective of this second part is therefore to use techniques inherent to Reinforcement Learning to develop a model that provides a distribution of assets for each user, according to their risk profile.

3.2 Data

Data from different S&P 500 stocks as well as US funds and ETFs have been used. The data has been taken from January 1, 2008 to June 10, 2022 and splitted for the training. The data has been normalized with minmax progressively to avoid looking into the future bias.

3.3 Environment

In Reinforcement Learning, the environment is the representation of our problem and the scenario with which the agent will act, from which it will receive the data and to which it will apply certain actions. For the realization of this Environment, the base Environment provided by the RLlib framework has been taken as an example and has been modified to adapt it to the requirements of this project. The environment used follows the structure of the OpenAI Gym framework environments (Brockman et al.

2016 [3]) and therefore requires certain functions: *init*, *step*, and *reset*, and certain components such as *state*, *reward* and *action*.

3.3.1 State

The state is what the agent analyzes and uses to take an action. Every time the agent performs that action affects the state that is passed. again to the agent to take a new action and so on. In this case, the status includes the covariance matrix of the returns of the financial instruments in the last year (252 days), technical data of each instrument for each day, the turbulence of the markets, the maximum level of turbulence supported by the user and the current weights of each instrument in the portfolio.

3.3.2 Initialization

When the environment is initialized, all the variables are initialized, and the *actionspace* and the *observationspace* are declared. The action space is the dimensionality and type of the actions that can be applied to the environment, and it is an array of n values ranging from 0 to 1 for $n =$ number of assets (stocks, bonds, ETFs) available. The observation space is the dimensionality and the type of data that the state will return after each action and that will be later interpreted by the agent. It is an mxn matrix for $n =$ number of assets and $m = n +$ number of technical indicators $+ 2$ (array of turbulence and current portfolio weights). In this step, the weights of each instrument in the portfolio are initialized. All the weights have a value of 0 at the beginning except for the *cash* variable whose weight is 1 at the beginning since it is assumed that the user has all his money in the account and without investing. The current value of the portfolio and some memory variables in which the different actions, values and returns will be stored are also initialized. Finally, the state of the environment is initialized.

3.3.3 Step

The step function is the core function of the environment because it is where things really happen. The function receives an action as a parameter. It first checks if it has reached the end of the dataset and therefore is in terminal state. In case that it is a terminal state, the function will return relevant information such as total profits or daily returns. In case that the function is not in a terminal state, it will proceed as follows. It will first normalize the action to ensure that the sum of all the weights is 1. Then it will store the weights in the memory of the environment. Subsequently, it will advance by n days, being n the period for rebalancing the portfolio's weights. During this project $n = 1$ has been chosen and therefore the agent analyzes and changes the weights every day. However, due to the way the environment is configured, it would be easy to change to other values if we want an agent that acts quarterly, for example, or on the contrary, it is a more active agent that acts several times a day. Next, the covariance matrix of the returns of the last year until the day in question is taken and updated in the state. When added to the state, the covariance values are scaled to keep them on a similar scale to the rest of the values. The new updated weights are also added to the state. Subsequently, the necessary elements for the reward function are calculated. This part will be explained in a more detailed way in the next section.

3.3.4 Reward function

The reward function is the function that returns feedback to the agent to indicate how well it is doing the required task. In general, the value of the reward can be positive or negative, and the agent will always try to maximize it. It is not easy to design a good and suitable reward function, since any imprecision can result in the agent learning the wrong strategies. This process has taken considerable time during the elaboration of this project and various functions have been tested along the way. There are different ways to measure the risk of a portfolio, the most common of them is its volatility but there are more. The challenge has been to find a way to reward returns while punishing losses that exceed a certain limit. Finally the best way to achieve it was thanks to the concept of financial turbulence. The financial turbulence index was introduced by Chow et al. (1999) [6] and was explained and extended in detail by Kritzman and Li (2010) [16]. As Kritzman and Li explain, the measure of financial turbulence measures the statistical unusualness of a set of returns given their historical pattern of behavior. This calculation coincides with the Mahalanobbis distance calculation, and its value is useful in calculating the amount of risk needed to earn returns. Kritzman and Li showed that in periods of great crises such as the Gulf War, the explosion of the tech bubble or the global financial crisis of 2009, the turbulence values skyrocketed. In addition, they also showed that turbulence persists, and therefore once it has risen, it remains high for a while, which is useful for predicting moments of low profitability. These findings served as the inspiration for the following final reward function.

$$r = P - (P * |c - t| * I) \tag{3.1}$$

Here the value P is the value of the portfolio. $|c-t|$ is the absolute difference between the weight of the cash and the percentile in which the turbulence value is with respect to past values (to avoid any look-ahead bias). The idea behind this is that the percentage of cash should be inversely proportional to the level of financial turbulence at the time, since when turbulence is high is when markets crash and when the value is low markets are calm and can rise. However, this strategy can sacrifice returns, since all the money that is in cash will not be able to grow. Therefore, there is an activation parameter of this strategy, the indicator I. The I parameter can only be 1 or 0. If the current turbulence level is greater than the turbulence threshold of the user, that is the maximum level of financial turbulence supported by the user calculated based on their risk profile, then $I = 1$. Otherwise $I = 0$ and the reward is the current value of the portfolio. In this way, medium and long-term strategies are only sacrificed if the situation becomes too risky for the user.

3.3.5 Reset

The reset function is the one that resets the values of the environment at the end of each training episode or whenever it is required. Most values return to their initial state. If the environment is run for a training process, the turbulence threshold and the initial day are reset with random values. In this way more randomness is added and the model is prevented from learning some wrong and anecdotic patterns.

3.4 Actor - Critic algorithms

The actor-critic algorithms are Reinforcement Learning algorithms that are characterized by having two models inside: the *actor* and the *critic*. As its name suggests, the actor is the model that is responsible for analyzing the current state of the environment and returning an action. The critical player instead takes that action and assigns it a score based on its suitability for the current state. These types of algorithms are compatible with actions in a continuous space, as our problem requires, since the actions in this case are equivalent to the weights of each element of our portfolio. This phase of the experimentation has been carried out with three models of the actor-critic type and that have certain differences between them. The models are: Proximal Policy Optimization (PPO), Twin Delayed DDPG (TD3) and Soft Actor Critic (SAC). These models have been trained using the $stable - baselines3$ library that contains different Reinforcement Learning algorithms already preconfigured and ready to be trained and tested. It is also already integrated in the FinRL library. Initially, the 3 models were trained with a total number of 300,000 steps each. The results showed that PPO was the most promising algorithm for this problem, so it was retrained, this time with 1,000,000 steps, and its hyperparameters were tuned, mainly the learning rate, which finally has a value of 0.0001, and the size of the batch, which is 128. The main characteristic of PPO is the use of memory batches that it uses to update the policy. After the training, the model has been tested putting it in comparison with a benchmark, which in this case has been the Dow Jones 30 since it is tested with shares of the Dow Jones 30..

3.5 Results

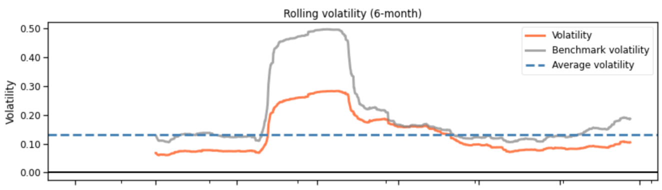
The results have been satisfactory. It can be seen how the algorithm employs more or less risky strategies depending on the maximum level of turbulence tooled by the user, and that even with the highest possible tolerance level, it manages to lighten the falls compared to the benchmark. It is interesting to see how this happens even during the unexpected crash of the markets caused by COVID-19, in which the agent manages to stop the fall in any scenario. The different results can be seen in the following figures.

Start date	**2019-01-02**
End date	**2022-06-09**
Total months	**41**
	Backtest
Annual return	9.635%
Cumulative returns	37.23%
Annual volatility	14.223%
Sharpe ratio	0.72
Calmar ratio	0.56
Stability	0.90
Max drawdown	-17.311%
Omega ratio	1.15
Sortino ratio	1.03
Skew	-0.14
Kurtosis	10.22
Tail ratio	0.94
Daily value at risk	-1.751%
Alpha	0.03
Beta	0.56

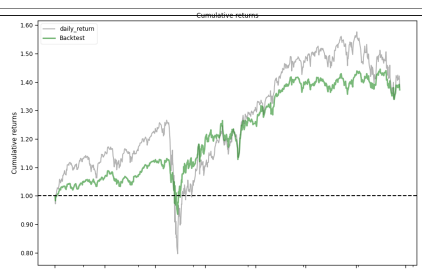
(a)

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	17.31	2020-02-13	2020-03-23	2020-05-11	63
1	10.79	2020-09-02	2020-10-28	2020-11-16	54
2	7.34	2022-03-30	2022-05-20	NaT	NaN
3	5.23	2020-06-08	2020-06-11	2020-07-01	18
4	5.06	2021-11-08	2021-12-01	2022-01-04	42

(b)



(c)



(d)

Figure 3.1: Results of the ppo algorithm for 1%turbulence tolerance

4 Graphical interface

The graphical interface has been created with the *streamlit* framework, which allows the entire webapp creation process to be carried out in python. It consists of a single page and its operation is simple. There is a form that the user must fill out to obtain their risk score. Once obtained, the page returns an investment strategy according to their risk profile using the previously trained Reinforcement Learning model.

5 Conclusions

In conclusion, this research project has shown that it is possible to automate risk profiling and asset allocation processes through the use of machine learning from start to finish. It has begun by finding and polishing relevant data to be able to segment the population into different risk profiles. To achieve this, numerous clustering algorithms have been used, as well as different metrics for cluster validation and techniques to tune the hyperparameters of each algorithm. Finally, the spectral clustering method has turned out to be the most effective for the data with which we worked. Once the clusters have been obtained, a further step has been added to personalize the process using the z-scores. Subsequently, different Reinforcement Learning agents have been trained in an environment specifically created to train agents who not only prioritize the return on investment, but also take into account the tolerance and risk capacity of each individual. The results of the agents have then been compared with a reference benchmark such as the Dow Jones 30 and it has been shown that the Proximal Policy Optimization (PPO) algorithm performs much better when it comes to protecting the user against risk taking into account the user's risk profile and without having to sacrifice returns. Finally, a small but practical application has been created so that these models can be accessible to the public.

6
Future improvements

Although full automation of risk profiling and asset allocation processes has been shown to be possible, the methods described here are far from complete and ready for the real world. The risk profiling would improve substantially with more adequate data to be able to train the models. For example, it would be very helpful if you could have a complete history of all the client's past investments, to see how they have operated in certain situations with high financial turbulence in the markets, and if they have learned to adapt their strategies over time. This part of the process would also improve with an evaluation of the clusters by experts in various sectors such as finance, risk, psychology and others, who are capable of better relating the traits of the members of each cluster with a specific attitude towards risk. Regarding the automation of the asset allocation process, this can be expanded to other types of products such as futures, options, currency exchanges in the Forex market or any other type of investment.

Bibliography

[1] *sklearn.cluster.optics*.

[2] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander, *Optics*, **28**, no. 2, 49–60 (american).

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba, *Openai gym*, arXiv preprint arXiv:1606.01540 (2016).

[4] T. Calinski and J. Harabasz, *A dendrite method for cluster analysis*, **3**, no. 1, 1–27 (american).

[5] Ricardo J. G. B. Campello, Peer Kröger, Jörg Sander, and Arthur Zimek, *Density-based clustering*, **10**, no. 2 (american).

[6] George Chow, Eric Jacquier, Mark Kritzman, and Kenneth Lowry, *Optimal portfolios in good times and bad*, Financial Analysts Journal **55** (1999), no. 3, 65–73.

[7] D. Comaniciu and P. Meer, *Mean shift: a robust approach toward feature space analysis*, **24**, no. 5, 603–619 (american).

[8] David L. Davies and Donald W. Bouldin, *A cluster separation measure*, **PAMI**, no. 2, 224–227 (american).

[9] John E. Grable, *Financial risk tolerance: A psychometric review* (american).

[10] Mao Guan and Xiao-Yang Liu, *Explainable deep reinforcement learning for portfolio management: An empirical approach* (american).

[11] Aurélien Géron, *Hands-on machine learning with scikit-learn, keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems*, 2 ed., O'Reilly Media (american).

[12] M. Halkidi and M. Vazirgiannis, *Clustering validity assessment: finding the optimal partitioning of a data set* (american).

[13] Terrence A. Hallahan, Robert W. Faff, and Michael D. McKenzie, *An empirical investigation of personal financial risk tolerance*, **13**, 57–78 (american).

[14] Zhengyao Jiang, Dixing Xu, and Jinjun Liang, *A deep reinforcement learning framework for the financial portfolio management problem*, arXiv preprint arXiv:1706.10059 (2017).

[15] Joachim Klement, *Investor risk profiling: An overview* (american).

[16] Mark Kritzman and Yuanzhen Li, *Skulls, financial turbulence, and risk management*, Financial Analysts Journal **66** (2010), no. 5, 30–41.

[17] Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Wang, *Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance* (american).

[18] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, Junjie Wu, and Sen Wu, *Understanding and enhancement of internal clustering validation measures*, **43**, no. 3, 982–994 (american).

[19] Nadia Rahmah and Imas Sukaesih Sitanggang, *Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra*, **31**, 012012 (american).

[20] Peter J. Rousseeuw, *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis*, **20**, 53–65 (american).

[21] Jörg Sander, Martin Ester, Hans-Peter Krieger, and Xiaowei Xu, *Density-based clustering in spatial databases: The algorithm gdbscan and its applications.*, **2**, no. 2, 169–194 (american).

[22] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu, *Dbscan revisited, revisited*, **42**, no. 3, 1–21 (american).

[23] Jaimie Sung and Sherman Hanna, *Factors related to risk tolerance* (american).

[24] Richard Sutton and Andrew Barto, *Reinforcement learning, second edition*, Amsterdam University Press (english).

[25] Laurens Van der Maaten and Geoffrey Hinton, *Visualizing data using t-sne.*, Journal of machine learning research **9** (2008), no. 11.

[26] Ulrike Von Luxburg, *A tutorial on spectral clustering*, Statistics and computing **17** (2007), no. 4, 395–416.

[27] Lihi Zelnik-Manor and Pietro Perona, *Self-tuning spectral clustering*, Advances in neural information processing systems **17** (2004).

[28] Tian Zhang, Raghu Ramakrishnan, and Miron Livny, *Birch* (american).

[29] Zihao Zhang, Stefan Zohren, and Stephen Roberts, *Deep learning for portfolio optimization*, **2**, no. 4, 8–20 (american).