GROUP WORK PROJECT # 1
GROUP NUMBER: 6200

MScFE 632: Machine Learning in Finance

FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Boyan Davidov	Bulgaria	davidovg@abv.bg	
Ivan Shigolakov	Russia	Shigolakov@yandex.ru	
Bharat Swami	India	bharatswami1299@gmail.com	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).	
Team member 1	
Team member 2	
Team member 3	

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed. **Note:** You may be required to provide proof of your outreach to non-contributing members upon request.

Classification Trees

Basics:

Classification tree is a map of binary decisions leading to a decision about a class or label. In other words, it is a series of if-then rules.

Keywords:

Decision tree, classification tree, DecisionTreeClassifier

Advantages:

- The classification trees models have a common feature: their results can be easily explained to other people (non-experts).
- The classification tree can be plotted and hence easily interpreted.
- Another good feature is that such models can divide the data non-linearly when searching for best trees (in comparison with some linear models).
- The model can use qualitative features without, say , one-hot-encoding or ordinary-encoding the features.

Computation:

For simplicity we will use a dataset about diagnosis with several features (Kaggle).

Disadvantages:

- The predictive power of classification trees is not so good in comparison to other classification methods
- These model can be very non-robust

Equations:

The algorithm is based on two steps:

1. Division of a feature space $X = \{X1, X2, X3, \ldots Xp\}$ into n different and non-overlapping small sets $R_1, R_2, R_3, \ldots, R_n$
2. Splitting the space so that the cost function is minimized: for classification tree we are using the Gini impurity or entropy which evaluation the quality of each split:

Gini impurity: $G = 1 - \sum_k (p_k)^2$

Entropy: $D = -\sum_{k=1}^{K} p_{mk} log(p_{mk})$

For each split the predicted response is the most commonly occuring class.

Features:

- The model can be used to get only qualitative results
- Resembles the human decision-making process (reasoning, logic)
- Ability to handle numerical and categorical data
- Dealing with missing values and nonlinear relationships
- Robust to outliers

Guide:

- The classification tree model takes two arrays as inputs:
    - An array X (training set or predictors) with the size of (N samples, M features)
    - Array Y (labels for training set) with the size of N samples
- The output of the model:
    - Decision tree, probability array

Hyperparameters:

- Gini index, entropy (information gain) (Criteria)
- Maximum depth the tree can grow (max_depth)
- Minimum number of samples for node-splitting needs (min_samples_split)
- Minimum number of samples at a leaf node (min_samples_leaf)
- Number of features to consider when searching for the best split (max_features)
- Minimum fraction of input samples required at a leaf node (min_weight_fraction_leaf)

Illustration:

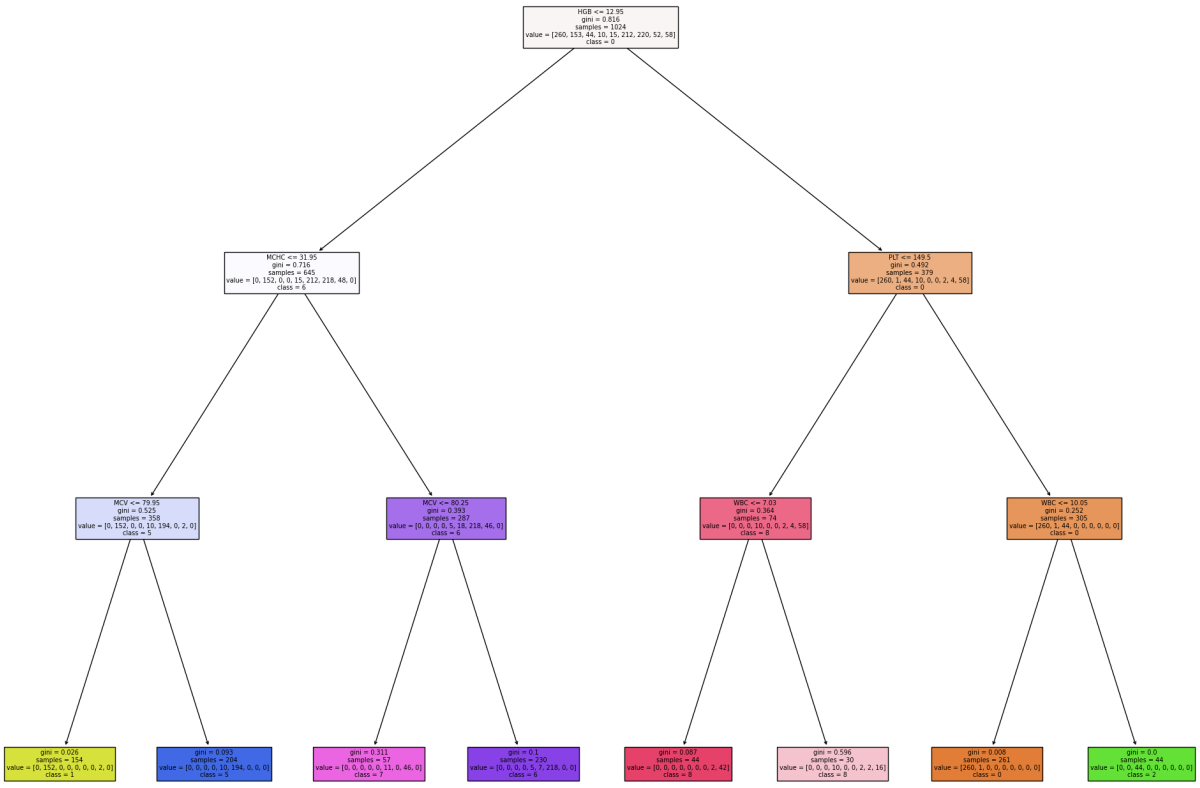

Figure 1

Journal:

Alexandre Momparler, Pedro Carmona & Francisco Climent (2016): Banking failure prediction: a boosting classification tree approach, Spanish Journal of Finance and Accounting / Revista Española de Financiación y Contabilidad:

K-Means Clustering

Basics

K-means clustering is an unsupervised machine learning technique where the goal is to group in a sensible way (considering similarities) individual data observations. Each data point is presented as a vector based on the features assigned to it. The data points don't need to be labeled (i.e. classified) in advance, hence the unsupervised notion of the method.

Keywords :

unsupervisedlearning, scree plot, centroids, Silhouette Analysis

Advantages:

- Efficiency: K-means is better than other clustering algorithms with its complexity of $O(nkt)$where n is the number of observations, k is the number of clusters and t is iterations until convergence. This is much more efficient than Hierarchical clustering with its complexity of $O(n^3)$
- Simple Algorithm easy to code: the idea is to guess the centers of each cluster and then iterate until convergence
- Flexible: Works well on different datasets regardless of size or type of data (financial, climate, etc.)

Computation:

Computation part is in the attached colab notebook

Disadvantages:

- First of all,we have to pre-set the number of clusters to be created. There are different techniques to find out the optimal number of clusters, yet it is an additional step to consider. Sometimes the number of clusters needed is obvious based on the nature of the data. If we are lucky it will be easy to conclude it just by using the elbow method. In the end it is also possible that some clusters are quite large whereas some other clusters include only a few data points (or even 1)
- Outliers can mess up the final output. Scaling and deleting outliers is advisable.
- Running the procedure several times can produce different results simply because the initial center of clusters is chosen randomly.

Equations:

Equations that summarize how the model works Mathematically, the objective of the K-means algorithm is to minimize an objective function which in our case is a squared error function given by

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||X^{j}_{i} - C_{j}||^{2}$$

where

$$||X^{j}_{i} - C_{j}||^{2}$$

is the Euclidean distance between data points x_{i} and the centroid C_{j} .

Features:

- It can handle high-dimensional data (eg. many features) and can perform relatively good in problems where other methods suffer from the curse of dimensionality
- It might not be suitable for time-series data because the timestamp will not play a role in the model

Guide:

- Input: * Number of clusters * The features matrix X
- Output: * Centroids of the clusters * Classification for each datapoint (label)

Hyperparameters:

List of hyperparameters that need tuning-

- Number of clusters
- Initialization method (random, k-means++)
- Number of iteration

Illustration: Visuals (figures, flowcharts, graphs) that show HOW the model works;

Below diagrams illustrates the k-mean clustering

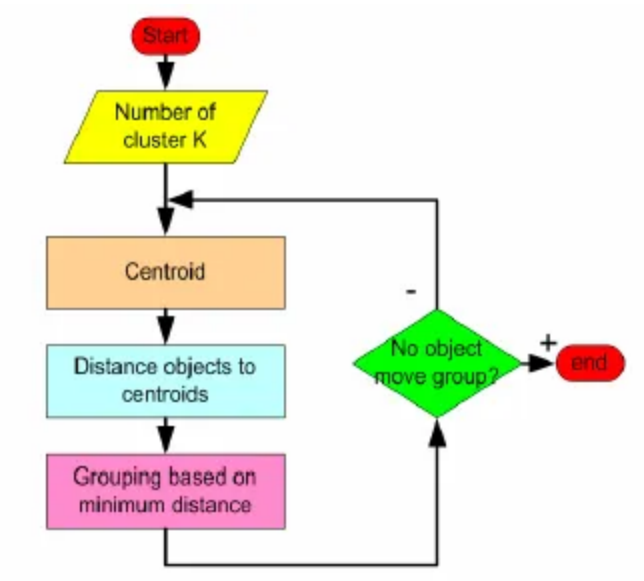

Figure 2

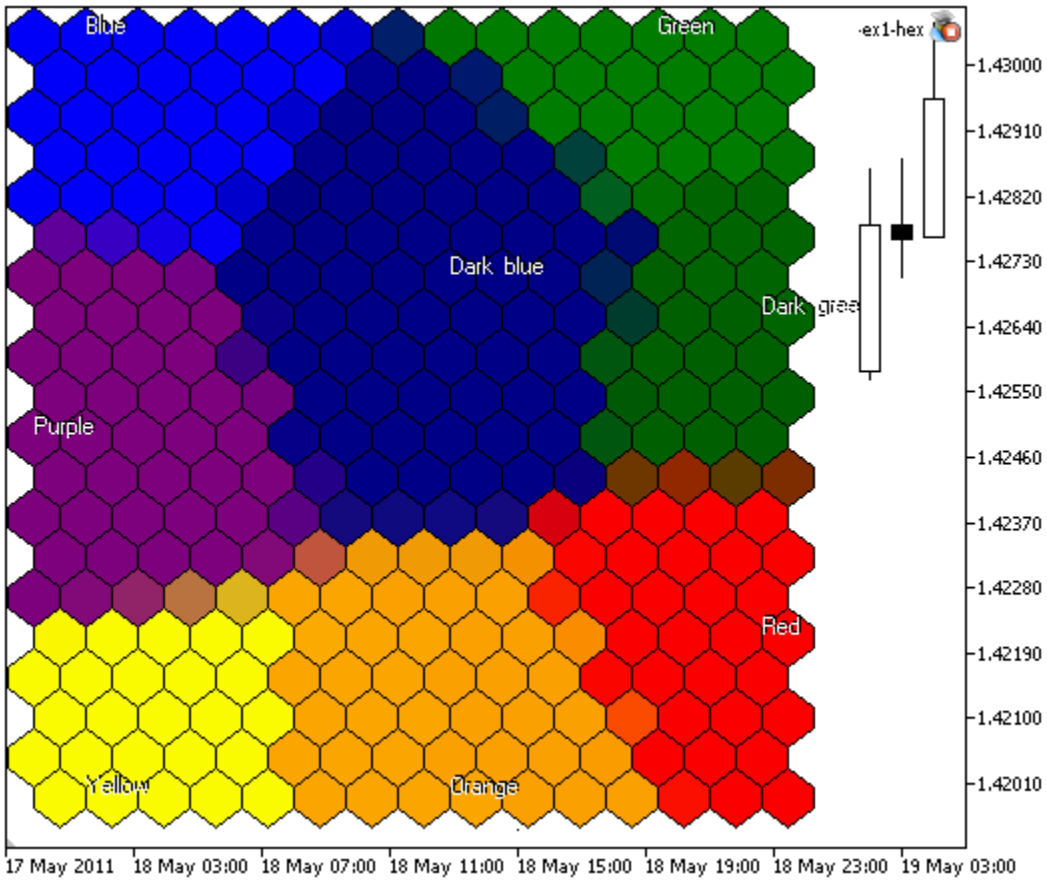

Figure 3

Journal:

https://www.researchgate.net/publication/379664052_Classifying_and_Analyzing_Physical_Activities_through_Heart_Rate_Variability_and_Other_Physical_Metrics_Using_Holter_Monitor_Data Authors: Boyan Davidov, Boyan Markov, Simona Mircheva

This is a paper that discusses different clustering methods beyond k-means and actually indicates that the curse of dimensionality can be also an issue for k-means clustering when there are many features (52 features). Indeed k-means is good for big datasets but not huge. Methods like t-sne (t distributed stochastic neighbour embedding) and UMAP (Uniform Manifold Approximation and Projection) can solve issues with huge datasets when we have millions of data points.

Principal Component Analysis (PCA)

Basics:

PCA is an unsupervised machine learning algorithm which helps in reducing the dimension of the dataset. Data Analysis becomes very difficult when we have a large number of features for every instance in the dataset, this is called the "Curse of Dimensionality", which is a huge problem since it increases the complexity of the dataset and needs more computational power. PCA helps in reducing the dimension of the dataset by choosing the weighted linear combination of original features which corresponds to the variance in the dataset by some accepted degree.

Keywords:

PCA, Unsupervised Learning, SVD, Eigenvalues, Eigen vectors, Principal Components, Dimension Reduction

Advantages:

Below are the advantages of using PCA-

- Dimension reduction: PCA helps in reducing the dimension of the dataset by taking features which explains the most of the variance in the dataset.
- Computational Cost: PCA reduces the computational cost of the model by reducing the dimension of the dataset.
- Data Visualization: PCA helps in visualizing the data. By reducing the dimension of dataset to 2D or 3D it helps visualize the high dimensional data up-to some accuracy (depends upon PCA preservation parameter)
- Reduces the Overfitting: By reducing the features of the dataset, PCA helps in reducing the overfitting problem.

Computation

Computation part is in the attached colab notebook.

Disadvantages:

PCA helps with the most common problem of data analysis which is "Curse of Dimensionality", but it also have some disadvantages, which are -

- Linear Assumption: PCA assumes that the calculated principal components are in linear combinations of original features, which may ignore the non-linear or more complex combinations.

- Standardization required: PCA involves an intermediate step of making data standard or normalize with mean 0 and variance 1. PCA is sensitive to the standardization process. Without this step PCA leads to some incorrect results.
- Interpretation: PCA is hard to interpret since they are linear combinations of some original features which hold the highest variances.
- Loss of Information: By reducing the dimension, PCA definitely losses the information.
- Computational Cost: If the dataset is huge, PCA requires high computational power and memory to run the model.
- If the number of instances in the dataset are less than the features, it is generally recommended not to use the PCA because it loses information.

Equations:

Below are the equation used in PCA

$X: \ Original\ Dataset\ with\ mxn$

now, Standardization Step

$B = X - \overline{X}$, where $\overline{X}$ is the mean of each feature in the data set.

SVD step

$B = U\Sigma V^T$ where U and V are unitary matrices with dimensions nxm and mxm, and Σ is nxm diagonal matrix with first m rows have diagonal values and then all row are zero.

PCA step

$T = BV$ where T are Principal components and V are working as Loading Eigenvectors of B.

so,

$T = BV = U\Sigma$ these are the principal components of our data set X.

➔ **we can use a alternative way to calculate the calculate the PCA which is as following-**

Standardization step : similar as above

Calculating theCovariance Matrix:

$S = [\,[\,\sigma_{11}\ \sigma_{12} \dots\ \sigma_{1n}],$

$[\,\sigma_{21}\ \sigma_{22} \dots\ \sigma_{2n}],$

$[\,\ldots\ldots\ ],$

$$[\sigma_{n1} \ \sigma_{n2} \dots \ \sigma_{nn}]]$$

Calculating the EigenValues:

$det(S - \lambda I) = 0$, this eigen equation will give us eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_m$.

Calculating the EigenVectors:

$$(S - \lambda I)U = 0$$

$$e_1 = \frac{U}{||U||} \ where \ ||U|| \ is \ determinant \ of \ Matrix \ U \ and \ e_1 \ is \ first \ eigenvector$$

Calculating the Principal components

$$e_j^T (X_j - \overline{X}) \ where \ j \ is \ from \ 1 \ to \ m$$

To choose the kth Principal components we have to choose kth largest eigenvalues. So, for the first principal component we have to choose the largest eigenvalue.

Features:

Below are the features of the PCA:

- PCA reduces the dimension of the dataset which helps in understanding the dataset by looking at the main features which show the highest variance.
- PCA are good with large datasets but if the dataset is very large it may cost more computational power and memory. To tackle this challenge we can use IPCA (Implemented PCA) which splits the large dataset into mini-batches to fit the available memory space. We can also use Randomized PCA which uses the stochastic algorithm "randomized PCA '' that quickly finds an approximation of d (dimension from PCA which preserves the required variance).
- PCA uses the SVD method and is useful to understand the SVD at greater extent for some dataset matrices.
- We can perform Inverse Transformation to retrieve the original data from the PCA dataset and since the PCA does not preserve the data completely this transformation data may not be completely same as original unless we use all the Principal components in the first place.

Guide

- Input
    - X : Original Dataset with m instances and n features => X_{mxn}

  - d : reduced dimension
  - Percentage preserve : we can use percentage preserve which states to the model to preserve the data up-to this percentage. Alternative to the d parameter.
- Output:
  - List of Principal components with their linear combinations of original features.
  - Loading for each feature for each principal component.
  - Percentage of variance explained by each principal component.

Hyperparameters

Below are the hyperparameters that need tuning in PCA:

- d : dimension, PCA doesn't know up-to which point we need the principal components to preserve data, PCA automatically gives n Principal components where n is the number of original features in our dataset by preserving all the variance in the dataset. we can tune the d hyperparameter according to over need and which can be done by the help of Elbow curve.
- We can tune the hyperparameter which is responsible for using the different PCA methods under the SVD method. For Example, Randomized PCA, Implemented PCA,etc.
- We can also tune the hyperparameter which if valued to True results in giving the principal components with the unit variance.

Illustration

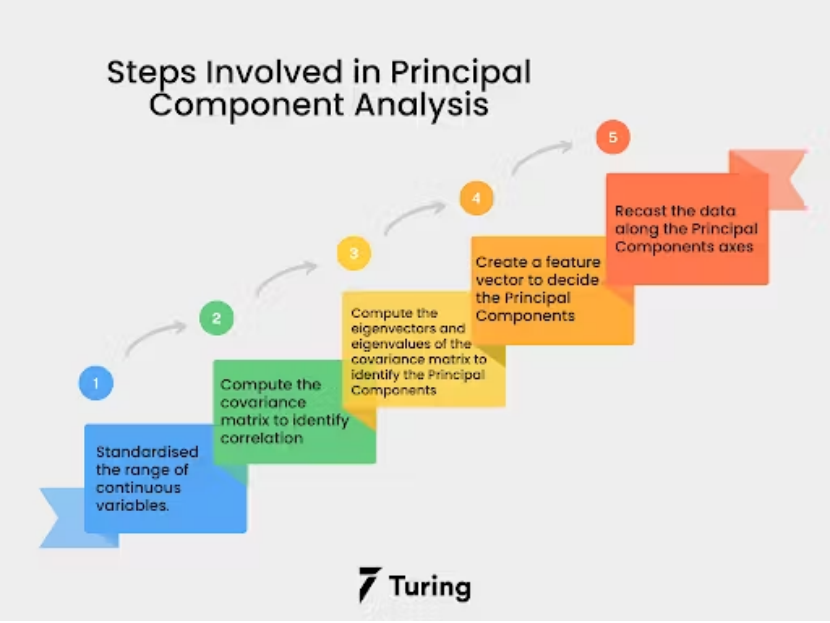

Figure 4

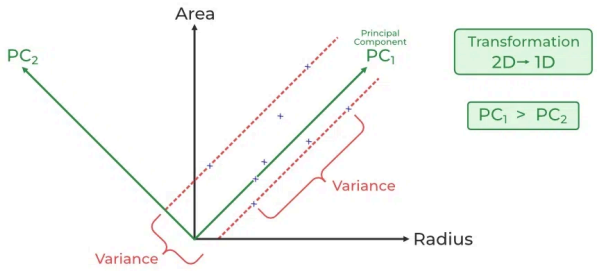

Figure 5

Journal:

Ian T. Jolliffe and Jorge Cadima, (Jan 2016) , Principal component analysis: a review and recent developments.

This article summarizes the PCA in a very subjective manner which is very easy for a layman to understand and use. This also talks about new development in the area of PCA research, since its 2016 published article these are not that much new but still helps to understand the PCA model.

Technical Section

The classification tree model has several hyperparameters.

By changing the hyperparameters (finding the best set) we can obtain better results in searching for a good model.

For example, we can sequentially change the maximum depth the tree may grow (max_depth).

There are three widely-used methods in searching the optimal hyperparameters:

- Grid search
- Random search
- Bayesian optimization

In our case we will use the Grid search method.

From the PCA model, we can see the d hyperparameter plays a very significant role in the algorithm. If we choose the d very large then we have high preservation of over original data variance but computational cost also increased since the PCA has time complexity of $O(mxd^2) + O(d^3)$, as we can see d plays a very crucial role in time complexity.

And also we can't choose the d very low since we need to preserve out information as much as possible.

One great way to choose the d hyperparameter is by using the hyperparameter tuning method which is called "Elbow plot" which shows the d versus the SSD (sum of squared distance) and we can choose the d such that the SSD does not change significantly as we increase the d. It makes an Elbow like graph.

Another hyperparameter we can choose in PCA is whether we want the standard PCA or modified PCA for example Randomized PCA, Implemented PCA, etc. These helps the model to reduce the time and space complexity.

Marketing Alpha

The basic features and advantages:

Features:

The model can be used to get only qualitative results

- Resembles the human decision-making process (reasoning, logic)
- Ability to handle numerical and categorical data
- Dealing with missing values and nonlinear relationships
- Robust to outliers

Advantages:

The main advantage of the classification tree model is an ability to easily interpret its results. There are many fluids where the classification tree models can be used: in medicine to detect the disease of patients who have certain characteristics. In this case for example doctors definitely need well-interpreted results to make decisions as they don't have mathematical or statistical background. In the web to identify spam, in computer vision to identify characters, in unmanned vehicle systems to identify objects and so on.

As has been mentioned earlier the classification tree model can handle non-linear relationships in a dataset. And in comparison with linear regression models this feature definitely makes classification trees much more preferable.

While modeling classification trees we can obtain very simple models through pruning. This model will have generalization characteristics and at the same time will be robust. As a result with classification tree models having only several levels one can easily explain it to other people and integrate the obtained rules to some system for future usage.

PCA can be used in the market and creates the alpha for the market. The market has a very vast amount of data which contains lots of features for each instance which we generally call the “ Curse of Dimensionality” and which takes a lot of the time and money to interpret and extract useful information. PCA can help us here to reduce the features and provide the useful features which explains most of the information of the market (in particular the environment).

K-means clustering is also a very useful model to use in creating the alphas. We need the information related to different subdomains of the market which are interconnected to each other and affect each other the most. K-mean clustering can help us here by providing the clusters of similar market domains. This is the one high level example of k-means used for

creating the market alpha. We can also use it in low level calculations for individual submarkets in the market or economy.

Learn More

- "Decision Tree in Machine Learning". Geeksforgeeks. 15 Mar, 2024,
- Gaurav Dutta. "Hyperparameter Tuning in Decision Trees". Kaggle,
- "Decision Trees". Scikit-learn,
- Alexandre Momparler, Pedro Carmona & Francisco Climent (2016): Banking failure prediction: a boosting classification tree approach, Spanish Journal of Finance and Accounting / Revista Española de Financiación y Contabilidad,
- Agnieszka Strzelecka, Danuta Zawadzka. Application of classification and regression tree (CRT) analysis to identify the agricultural households at risk of financial exclusion. Koszalin University of Technology, Faculty of Economic Science, Department of Finance,
- Ian T. Jolliffe and Jorge Cadima, (Jan 2016) , Principal component analysis: a review and recent developments.
- https://www.researchgate.net/publication/379664052_Classifying_and_Analyzing_Physical_Activities_through_Heart_Rate_Variability_and_Other_Physical_Metrics_Using_Holter_Monitor_Data Authors: Boyan Davidov, Boyan Markov, Simona Mircheva

References:

- Figure 2: Source: [Haonan Wu, Jingyan Sun, Lingxiao Song, Ziyue Cheng, Learn K-Means and Hierarchical Clustering Algorithms in 15 minutes]
- Figure 3: Source: https://c.mql5.com/2/3/Figure5_som__1.png
- PCA part , Chapter 8 Dimensionality Reduction, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow , Author: Aurélien Géron, Third edition, 2023
- Figure 4 Source :[Dharani , https://www.turing.com/kb/guide-to-principal-component-analysis]
- Figure 5 Source [2023, Aishwarya, Principal Component Analysis(PCA) ,]
- Steve Brunton, Principal Component Analysis (PCA)
- Mahesh Huddar, 1 Principal Component Analysis | PCA | Dimensionality Reduction in Machine Learning by Mahesh Huddar
- Paul Wilmott: Machine Learning: An Applied Mathematics Introduction