Pages 1–8
2023

# Project 1 - To Higgs or not to Higgs

Bruno Parracho (202203990) and Telmo Monteiro (202308183)
MSc. in Physics, MSc. in Astronomy and Astrophysics
Mathematics Department, Faculty of Sciences of University of Porto

**Course**: Statistical Methods for Data Mining

**Course director:** Joaquim Fernando Pinto da Costa

**Keywords:** Machine Learning, Particle Physics, Statistical Methods, Data Mining

## ABSTRACT

The topic of this study is the analysis of a certain data set using statistical methods. Furthermore, it is concentrated on the binary classification using a large-scale data set for Higgs boson detection. Finding the ideal number of principle components for the assignment and examining the efficacy of principle Component Analysis (PCA) as a dimensionality reduction method are the main goals of the study. In order to make an informed decision, the validity of three commonly used criteria—the Kaiser, Pearson, and Cattell's criteria—is evaluated. In the end, the Kaiser criterion is the most effective approach, which results in the identification of 13 major components.

Afterwards, several clustering strategies are used to find patterns in the data. The first technique discussed is K-means clustering, which requires figuring out the optimal number of clusters. Three widely used approaches are used to accomplish this: the Elbow method, the Silhouette method, and the gap statistic. The best strategy for the Higgs detection job is determined to be the Silhouette method with K=2.

According to the preliminary clustering results, a distinct, non-overlapping cluster can be seen when comparing the First Principal Component (PC) with any of the other 12 PCs. But when the remaining PC combinations are examined, the clusters often overlap.

## CONTENTS



## 1 INTRODUCTION

### 1.1 Particle Physics

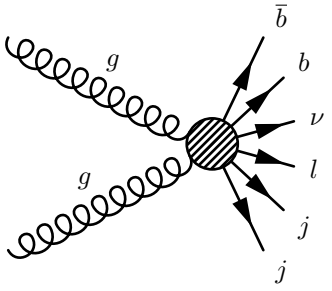
The goal of high energy physics, mostly known as particle physics, is to determine the fundamental building blocks of the universe. In order to achieve that goal, particle accelerators were built with the sole purpose of colliding particles with so much energy that leads to the creation of "new" particles. Among these particles is the world famous Higgs boson, detected in 2013 by the LHC at CERN. Detecting the Higgs boson was no easy task (to get a rough idea, there are $10^{11}$ collisions per hour and only 300) and it involves a good chunk of statistical analysis and machine learning tools to optimise the detection process.

### 1.2 Detection

To know if a particle was created or not, we need to compare the subspace of the data and the subspace of the null hypothesis (non creation of the Higgs) and check if the difference between the two is significantly big. Due to the fast decaying of particles, it is very tricky to detect the intermediate particles. Before that, lets define the final products which we can directly measure, looking at the processes that produce:

- 4 jets (quarks), of which 2 of the jets are heavy jets (b-quarks);
- 1 lepton (electron or muon) and 1 neutrino that will account for missing energy.
  Note that: Due to the nature of the neutrino it can't be directly measured and only indirectly via conservation laws.

The collision of these particles can be represented by the following Feynman diagram:




In a collider, we will have a background and signal process, and so the signal process is given by figure 1 and the background process is given by figure 2.

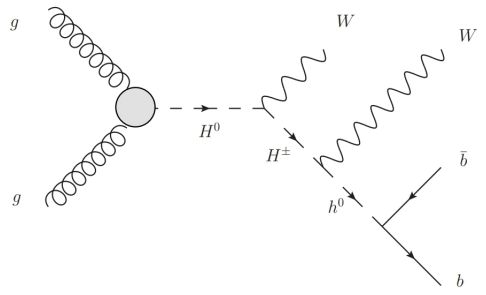



Figure 1: Signal Process. Extracted from Baldi et al. [2014].

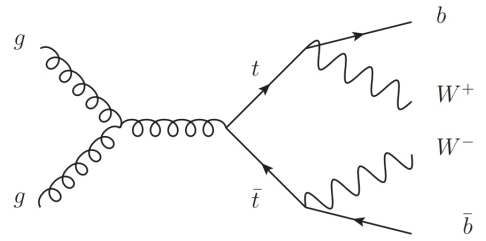



Figure 2: Background Process. Extracted from Baldi et al. [2014].

### 1.3 Input variables

The final product's properties we can detect are the transverse momentum, $p_T$, and 2 angles $\phi, \theta$ for the direction, where $\theta$ is rewritten as the pseudo-rapidity, $\eta = \tan(\theta/2)$. This way, we have 4 variables ($p_T, \phi\,\eta$, b-tag) for each jet and 3 variables ($p_T, \phi\,\eta$) for the lepton. We exclude the neutrino, of which we can only detect indirectly through the missing momentum and the angle $\phi$. In total, we have $(4 \times 4 + 1 \times 3 + 2) = 21$ low input variables. Another set of variables includes the mass distributions, consisting in 7 high input variables. This variables are considered high, due to their dependence on the occurring process.

## 2 DATA

The data set used in this work was produced by Baldi et al. [2014], containing 11 million simulated collision events for benchmarking machine-learning classification algorithms on this task. It can be found in the UCI Machine Learning Repository at `archive.ics.uci.edu/ml/datasets/HIGGS` (Whiteson [2014]).

The first 21 features (columns 2-22) are the kinematic properties measured by the particle detectors in the accelerator, as stated in the previous section. The last seven features are functions of the first 21 features: high-level features derived by physicists to help discriminate between the two classes. There is an interest in using deep learning methods to obviate the need for physicists to manually develop such features. Benchmark results using Bayesian Decision Trees from a standard physics package and 5-layer neural networks are presented in the original paper. The last 500 000 examples were used as a test set.

In the case of this work, we started by only analysing 10 000 randomly chosen events (the rows of the data set), so it could be computationally feasible.

Table 1 shows some basic statistics of the variables (or predictors) for a sample of 10 000 events of the data used, obtained using the "summary(data)" command in R. For the target column, whether the process results in a Higgs particle (1) of not (0), the result was 4639 for 0 and 5361 for 1.

## 3 PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal component analysis (PCA) is a technique in multivariate statistics that reduces the high dimensional space of variables in a given data-set of observations/measurements. It can accomplish this by finding an orthogonal basis where the data can be efficiently expressed, this is done by linearly transforming the original data, this basis has the property that most of the variance can be explain with the least possible number of basis vectors (Principal components) (Asensio Ramos et al. [2023]). This principal components can be expressed as the weighted sums of the original basis vectors, this is to say the principal components aren't specific initial variables but a combination of them.

To first analyse the data, we need to lower our high dimensional space, so we can facilitate our analysis and the methods done in the following sections. We used the built-in function in R, **prcomp**, to compute the principal components. It is also very important to note our data needed to be scaled, as in 1 we see different variables have very different ranges. As the 2 first principal components explain most of the variance, we can visualise how the initial variables correlate with each other by projecting the data onto a sub-space of the 2 PCs, and representing the variables as vectors. Using the built-in function in R, **fviz_pca_biplot**, we get a biplot in Figure 3

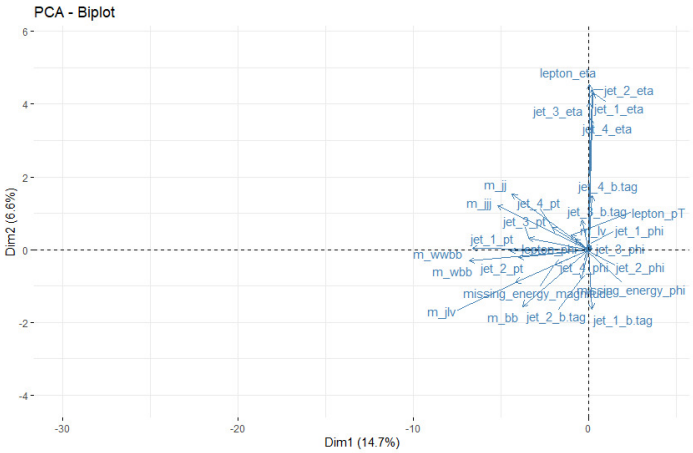



Figure 3: Biplot with the correlation between variables

As the principal components are linear combinations of the original variables, with different weights, it is important to keep in mind that if we remove some principal components, we lose information of our original data set.

There are some criteria to classify which principal components to consider, from which we'll consider the Kaiser-Guttman's, Pearson's and Catell's criterion.

- **Kaiser-Guttman Criterion**: The Kaiser-Guttman criterion states that we can remove the principal components for which the eigenvalues are below 1. Figure 4 shows the eigenvalues for all the principal components. In this case, the Kaiser-Guttman criterion will keep the 13 first principal components.
- **Pearson's Criterion**: The Pearson's criterion states that we can maintain the $q$ components that explain at least 80% of the variance. Figure 5 shows the percentage of variance explained by the principal components. In this case, the Pearson criterion will keep the 16 first principal components.
- **Cattell's Criterion**: The Cattell's criterion or elbow rule states that the eigenvalues of the components to be kept

Table 1: Some basic statistics about the data used.

| Feature | Minimum | 1st Quartile | Median | Mean | 3rd Quartile | Maximum |
|---|---|---|---|---|---|---|
| lepton_pT | 0.2747 | 0.5908 | 0.8655 | 1.0023 | 1.2509 | 5.7551 |
| lepton_eta | -2.431080 | -0.732722 | 0.006277 | 0.000139 | 0.725553 | 2.427076 |
| lepton_phi | -1.74195 | -0.82033 | 0.02844 | 0.02264 | 0.88376 | 1.74268 |
| missing_energy_magnitude | 0.0133 | 0.5662 | 0.8871 | 0.9958 | 1.2916 | 5.4989 |
| missing_energy_phi | -1.74390 | -0.88670 | -0.01693 | -0.01688 | 0.84569 | 1.74264 |
| jet_1_pt | 0.1945 | 0.6784 | 0.9020 | 0.9988 | 1.1796 | 5.7914 |
| jet_1_eta | -2.941998 | -0.673382 | 0.009382 | 0.009000 | 0.689175 | 2.943928 |
| jet_1_phi | -1.741237 | -0.853128 | 0.013466 | 0.001928 | 0.868451 | 1.741454 |
| jet_1_b.tag | 0.000 | 0.000 | 1.087 | 1.001 | 2.173 | 2.173 |
| jet_2_pt | 0.1891 | 0.6572 | 0.8912 | 1.0015 | 1.2086 | 6.0479 |
| jet_2_eta | -2.911147 | -0.687915 | 0.009774 | 0.003904 | 0.692650 | 2.902525 |
| jet_2_phi | -1.74237 | -0.89418 | -0.02018 | -0.01642 | 0.85670 | 1.74317 |
| jet_2_b.tag | 0.000 | 0.000 | 1.107 | 1.012 | 2.215 | 2.215 |
| jet_3_pt | 0.2636 | 0.6464 | 0.8854 | 0.9869 | 1.2124 | 6.3967 |
| jet_3_eta | -2.727842 | -0.696167 | 0.010186 | 0.007105 | 0.717449 | 2.726368 |
| jet_3_phi | -1.742069 | -0.878069 | 0.011832 | -0.001236 | 0.869730 | 1.742884 |
| jet_3_b.tag | 0.0000 | 0.0000 | 0.0000 | 0.9952 | 2.5482 | 2.5482 |
| jet_4_pt | 0.3654 | 0.6202 | 0.8777 | 0.9935 | 1.2303 | 8.8515 |
| jet_4_eta | -2.496432 | -0.710026 | 0.003703 | 0.009831 | 0.728260 | 2.492179 |
| jet_4_phi | -1.742691 | -0.875502 | -0.007201 | -0.005492 | 0.841085 | 1.743372 |
| jet_4_b.tag | 0.000 | 0.000 | 0.000 | 1.009 | 3.102 | 3.102 |
| m_jj | 0.1463 | 0.7887 | 0.8947 | 1.0435 | 1.0295 | 13.9386 |
| m_jjj | 0.3166 | 0.8504 | 0.9513 | 1.0324 | 1.0933 | 7.8016 |
| m_lv | 0.3295 | 0.9857 | 0.9898 | 1.0501 | 1.0190 | 4.9615 |
| m_jlv | 0.4063 | 0.7651 | 0.9204 | 1.0127 | 1.1478 | 9.7721 |
| m_bb | 0.07663 | 0.67552 | 0.87273 | 0.97943 | 1.14437 | 10.86245 |
| m_wbb | 0.3904 | 0.8232 | 0.9517 | 1.0407 | 1.1500 | 7.0254 |
| m_wwbb | 0.4206 | 0.7724 | 0.8752 | 0.9653 | 1.0682 | 5.4601 |

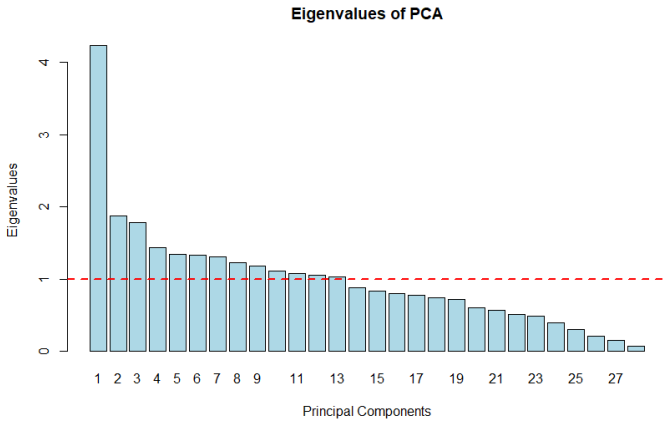



Figure 4: Eigenvalues of the principal components, with the horizontal line illustrating the cutoff of Kaiser-Guttman criterion.

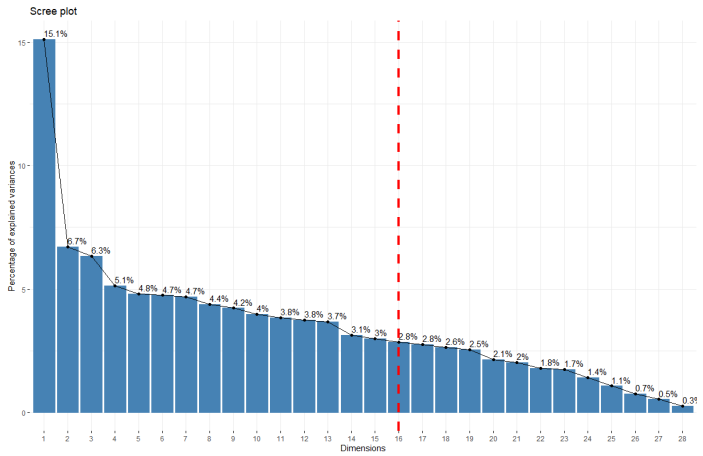



Figure 5: Percentage of variance explained by the principal components, with the vertical line illustrating the cutoff of the Pearson criterion.

need to have a small difference with respect to previous component, such that $\lambda_\alpha - \lambda_{\alpha-1} < \epsilon$. Looking at figure 4 again, there is no possible $\epsilon$ that keeps the first principal component and eliminates some of the others, so we opt to not use this criterion.

As we want to keep the minimum number of principal components possible, so we can facilitate the analysis, we opt for the Kaiser-Guttman criterion. Figure 6 shows a parallel coordinates plot for the 13 principal components that were kept with this criterion.

The correlation matrix of the original variables in the data is given by figure 7. The method used was the Pearson correlation coefficient. As one can see, the correlation gets more (positively) stronger in the high-level features (the mass distributions). The reason why this happens is beyond the scope of this work.

## 4 CLUSTERING

Clustering is a technique in machine learning that looks for the similar data points in the data set into groups. It is also sometimes referred to as a type of **unsupervised** machine learning task, in the sense that it doesn't take into account if the data points have an already predetermined outcome, labeled as **target**. So it doesn't necessarily mean this clustering will follow the desired outcomes.

In clustering, we have 2 types: hierarchical and non-hierarchical methods. The hierarchical methods look to intertwine the various clusters and the non-hierarchical methods look for partitions within the data set. The principal objective with this type of technique is to look for patterns between the variables.

### 4.1 Distance metrics

Before getting into clustering, we need to establish what distance metrics are. Data, or observables, are measured by different

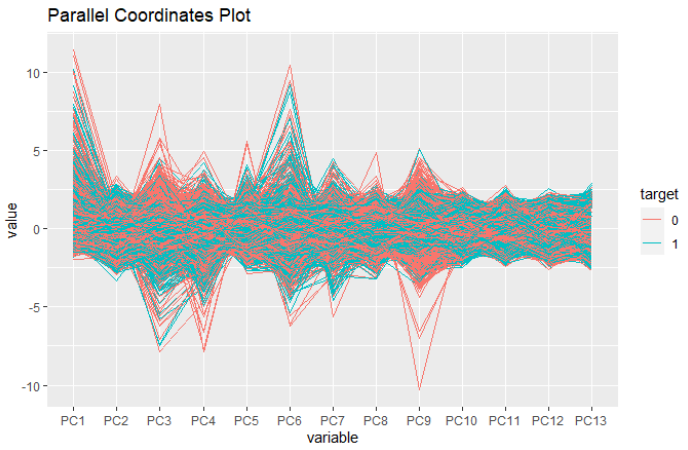



Figure 6: Parallel coordinates plot for the 13 principal components that were kept, colored according to if the event produces a Higgs or not.

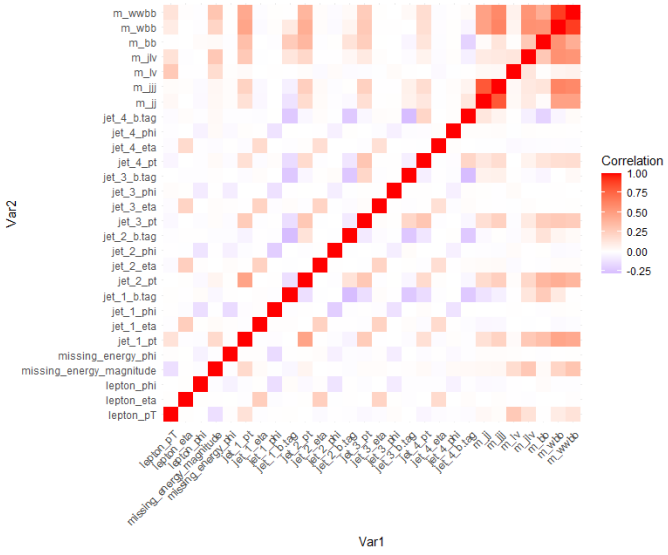



Figure 7: Correlation matrix of original variables using Pearson correlation coefficient.

distance metrics and grouped accordingly. To group the elements, one can use similarity or dissimilarity methods.

A distance between any pair of vectors or points $i, j, k$ satisfies the following properties of:

- Symmetry: $d(i,j) = d(j,i)$;
- Positive definiteness: $d(i,j) > 0$ and $d(i,j) = 0$ if $i = j$;
- Triangular inequality: $d(i,j) \leq d(i,k) + d(k,j)$.

If the triangular inequality is not taken into account, we have a **dissimilarity**. Finally, a **similarity** is given by $s(i,j) = \max_{i,j}\{d(i,j)\} - d(i,j)$.

The traditional way to measure distances is with the Minkowski distance, that generalizes a family of metrics defined as

$$L_p(\vec{x}_a, \vec{x}_b) = \left( \sum_{i=1}^{N} |\vec{x}_{i,a} - \vec{x}_{i,b}|^p \right)^{1/p} ; \forall p \geq 1, p \in \mathcal{Z}^+. \quad (1)$$

The Manhattan, Euclidean and Chebyshev distances are special cases of the Minkowski distance, with, respectively, $p = 1$, $p = 2$ and $p \to \infty$.

## 5 K-MEANS CLUSTERING

K-means is a technique in cluster analysis, that looks to represent $N$ data points into $k$ clusters, being a NP-complex problem. It is perhaps the most used technique of the partition-based methods and to minimize the K-means we need to use algorithms. In spite of that, there is no ideal algorithm, as it depends on the size, number of variables and composition of the data set. It also depends on the choice for the number of clusters, as we will explore later in this chapter.

The most widely used clustering algorithms do not take into account a probability model that describes the data. Instead, they immediately assign each observation to a group or cluster. A unique label for each observation is assigned by an integer $i \in \{1, ..., N\}$. It is assumed that there are a predetermined number of clusters $K < N$, each of which is identified by an integer $k \in \{1, ..., K\}$. Only one cluster is assigned to each observation. A many-to-one mapping, or encoder $k = C(i)$, that places the $i$-th observation in the $k$-th cluster can be used to describe these assignments.

Based on the differences $d(x_i, x_i)$ between each pair of observations, one searches for the specific encoder $C^*(i)$ that accomplishes the necessary goal. The user specifies these, as previously mentioned. In general, for each observation $i$, the encoder $C(i)$ is explicitly defined by providing its value (cluster assignment). Therefore, the unique cluster assignments for each of the $N$ observations serve as the procedure's "parameters." These are changed to minimize a "loss" function that represents the extent to which the clustering objective is not achieved (Hastie [2009]).

The approach we follow is to specify a mathematical loss function which we will look to minimize it. Therefore we will assign the close points to the same cluster, a natural loss function would be (Hastie [2009]),

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'}). \quad (2)$$

The search for the best clustering is based on associating to each partition a kind of error function associated with a sum over the squares of the deviations of each element to the center of the cluster. Using the euclidean metric, we can use Equation 2 to obtain

$$W(C) = \sum_{k=0}^{K} \sum_{C(i)=k} ||X_i - \overline{X}_k||^2 \quad (3)$$

where $C(k)$ refers to the K-cluster, and $||X_i - \overline{X}_k||$ is the Euclidean distance.

### 5.1 Algorithm

In order to minimize the loss function, we use the standard algorithm, known as naïve or Lloyd's algorithm. In spite of this algorithm being slow when compared to other alternative algorithms, its implementation is easy and direct. Given an initial set of K means $\{m_1^{(1)}, ..., m_K^{(1)}\}$, the algorithm proceeds by following the two phases:

1. A metric is used to assign each data point to its nearest centroid (mean).

$$S_i^{(t)} = \{x_p : ||x_p - m_i^{(t)}||^2 \leq ||x_p - m_j^{(t)}||^2, \forall j, 1 \leq j \leq K\} \tag{4}$$

where $(t)$ is the $t$-th iteration, the inequality represents the $i$-th cluster that is closest to $x_p$, and $S_i^{(t)}$ is the set of points belonging in the $i$-th cluster.

2. The means are updated based on the partition that was obtained in the previous phase.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \tag{5}$$

where $|S_i^{(t)}|$ is the number of points in $i$-th cluster at the $t$-th iteration.

When a predetermined maximum number of iterations is reached or no data point changes clusters, the iterative procedure comes to an end.

### 5.2 Validation methods

Before analysing the data using K-means, we need to define the optimal number of clusters and doing it is not a simple task. Many methods have been developed in order to do this task: direct methods and statistical methods.

Direct methods consist on optimizing some criteria, such as the average silhouette or the cluster sums of squares. These are respectively **silhouette** and **elbow** methods (Kassambara [2017]).

Statistical methods consist on comparing our data with a null hypothesis, for example with the **gap** statistic.

The methods in bold text are the ones we apply our K-means results to.

The built-in function in R, **fviz˙nbclust** is capable of computing this validation tests.

- **Elbow method**: the general objective is to minimize the loss function, defined in Equation 2. This method looks at the total loss function in terms of the number of clusters, K, and decides to choose a number of clusters such that the loss function doesn't change that much, in other words, $W(K) - W(K-1) < \epsilon$. Figure 8 shows how the total loss function varies with the number of clusters, and allows us to conclude the optimal number of cluster with this method is around $K = 4$ clusters.
- **Silhouette method**: as the elbow method is sometimes ambiguous (as seen in the PCA criterion section), we use an alternative that is the average silhouette method. This indicates how well the data inside a cluster fits in itself: a high average silhouette indicates a good clustering. The average silhouette width is given by $S = 1/K \sum_k s(C_k)$, where $s_k$ is the silhouette width for the observation $x_i$ and can be written as $s(x_i) = (b(x_i) - a(x_i))/\max b(x_i), a(x_i)$. Figure 9 shows us the average silhouette width with respect to the number of clusters. This shows us the optimal number of clusters is $K = 2$ clusters.
- **Gap statistic**: this statistic is an approach that can be applied to about any clustering method. It compares the total loss function variation for different numbers of clusters with their expected values under the null hypothesis distribution of the data. The maximum value is the one that optimizes the number of clusters. In other words, it means that the clustering configuration is far from the random uniform distribution of data (Tibshirani et al. [2001]). It also formalizes the elbow method, defining the function Gap as $\text{Gap}_n(k) = E_n^*\{\log(W(k))\} - \log(W(k))$, where $E_n^*$ is the expectation of the data under a reference distribution. In short, the smallest possible $k$ such that $\text{Gap}(k) \geq \text{Gap}(k+1) - s_{k+1}$ is the number of clusters we want to find. Figure 10 allows us to see the optimal number of clusters using this test is $K = 1$ cluster.

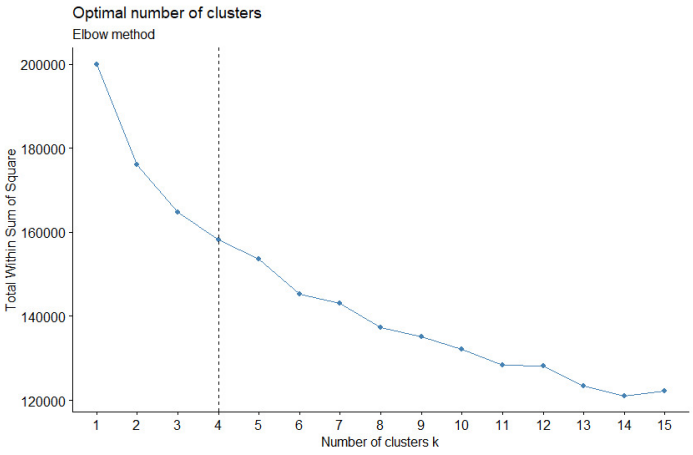



Figure 8: Elbow method: Total loss function in terms of the number of clusters

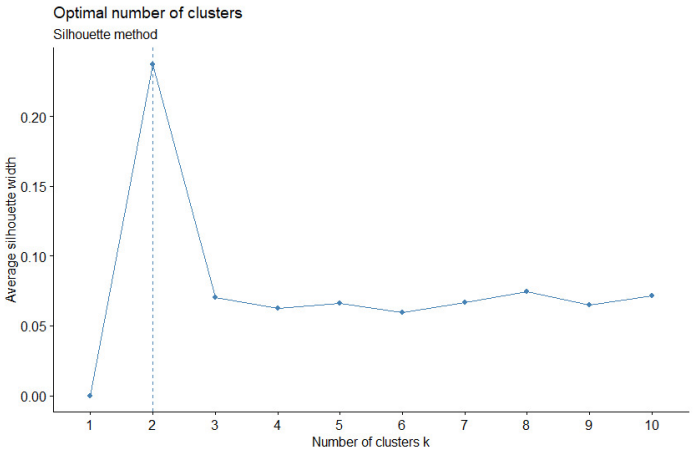



Figure 9: Silhouette method: Average silhouette width in terms of the number of clusters

There are then some factors to take into account, seeing the suggested number of clusters by each method:

- the initial problem consists of classifying an event in Higgs production or not (binary), so it is intuitive that we want to explore the clustering with 2 clusters;
- using $K = 1$ as suggested by the gap statistic is redundant, as the objective in clustering is to divide the data in more than 1 group. The second best $K$ that maximizes this statistic is $K = 2$.

This way, we have 2 methods that support $K = 2$ and 1 that supports $K = 4$. By majority and adding the first factor, we opted to use $K = 2$.

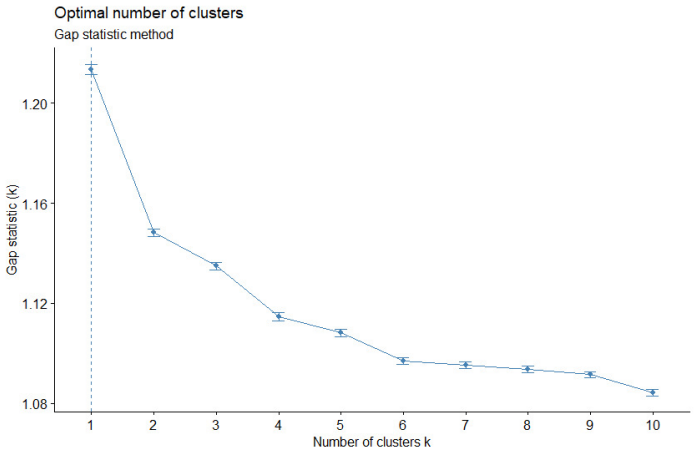



Figure 10: Gap statistic: Variation of the total loss function with the expected values of the data in terms of the number of clusters

### 5.3 Analysis

We used the R built-in-function **kmeans** to apply the standard algorithm. To graphically represent the clustered data according to the principal components analysed, we used the built-in function, **splom**. Figure 11 represents the clustering taking into account the different combinations of two principal components, from the thirteen principal components kept, as explained in the previous section. As one can observe, the first principal component PC1, when combined which any of the others, produces the most seemingly visible and separated clusters. This is expected, as the first principal component is the one that explains the most variance in the data set (more than double than PC2).

To inspect this result, we plotted using the function **fviz_cluster** the clusters with PC1 and PC2.

## 6 HIERARCHICAL CLUSTERING

As seen in the K-means clustering, it's defined a pre-defined K number of cluster and a set of means. In contrast, hierarchical clustering methods do not need such definitions. However, here we need to specify a measure of dissimilarity between the clusters of observations, based on the pairwise dissimilarities among the observations in the two clusters. As the name suggests, the clusters at each level of the hierarchy are created by merging clusters at the next lower level. In the lowest level, each cluster corresponds to each individual data point, and the highest level refers to the cluster that encloses every data point.

In hierarchical clustering, we can divide the strategies between 2 paradigms: agglomerative (bottom-up) and divisive (top-down). Agglomerative strategies starts at the lowest level and progressively merges a selected pair of clusters into a single cluster, this pair chosen is the one that has the smallest intergroup dissimilarity. By contrast divisive strategies only differ by starting at the highest level, and slipting into 2 clusters that produces the largest between-group dissmilarityHastie [2009].

The majority of agglomerative as the merger level increases, the differences between the merged clusters get monotonically bigger. As a result, the binary tree can be plotted with each node's height corresponding to the intergroup dissimilarity between its two resulting elements. Each individual observation's terminal node is plotted at zero height. We refer to this kind of graphical representation as a **dendrogram**.

### 6.1 Agglomerative

In this work we will use the R built-in-function, **hclust**, that uses a agglomerative approach.

## 7 NORMAL MIXTURE MODELS FOR CLUSTERING

The Gaussian mixture models (GMM) is a type of clustering that models the data as a mixture of various gaussian distributions. Unlike other approaches, GMM's probabilistic clustering approach gives each cluster a probability distribution, enabling more precise and adaptable grouping. GMM is capable of handling overlapping clusters and modeling intricate cluster shapes.

This model is able to search for a mixture of uni-modal gaussian distributions that better fit the data. The multi-modal density function is given by

$$p(X) = \sum_{i=1}^{K} \pi_i p_i(X) = \sum_{i=1}^{K} \pi_i N_p(\mu_i, \Sigma_i) \tag{6}$$

where $p_i = N_p(\mu_i, \Sigma_i)$, and $\pi_i$ is the weight of the $i$-th distribution, it's clear that this weights will follow $\sum_{i=1}^{K} \pi_i = 1$, and $K$ is the number of uni-modal distributions.

To find the best mixture of gaussian distributions, it's convenient to use the maximum likelihood estimator, that is a method to estimate the parameter space under a statistical model. So in order to obtain the best fitting of the model into the data set we need to maximize the likelihood. Therefore the likelihood can be given by,

$$\begin{aligned} &\mathcal{L}(\pi_1, ..., \pi_K, \mu_1, ..., \mu_K, \Sigma_1, ..., \Sigma_K, \vec{X}) \\ &= p(X_1, X_2, ..., X_N) \\ &= \Pi_{j=1}^{N} p(X_j) \end{aligned} \tag{7}$$

In order to maximize this method, we use the Lagrange multiplier,

$$J = \sum_{j=1}^{N} \ln(p(X_j)) - \lambda \left( \sum_{i=1}^{K} \pi_i - 1 \right) \tag{8}$$

where $p(X_j) = \sum_{i=1}^{K} \pi_i p_i(X_j)$, and $\lambda$ the Lagrange multiplier. Finding the extremes of this set of quantities $\left\{ \frac{\partial J}{\partial \pi_i}, \frac{\partial J}{\partial \mu_i}, \frac{\partial J}{\partial \Sigma_i} \right\}$ and setting a new probability called the posteriori probability, $q_i(X_j) = \frac{\pi_i p_i(X_j)}{p(X_j)}$, and solving each set we get

$$\begin{cases} \frac{\partial J}{\partial \pi_i} = 0 \\ \frac{\partial J}{\partial \mu_i} = 0 \\ \frac{\partial J}{\partial \Sigma_i} = 0 \end{cases} \Leftrightarrow \begin{cases} \pi_i = \frac{1}{N} \sum_{j=1}^{N} q_i(X_j) \\ \mu_i = \frac{1}{N\pi_i} \sum_{j=1}^{N} q_i(X_j) X_j \\ \Sigma_i = \frac{1}{N\pi_i} \sum_{j=1}^{N} q_i(X_j)(X_j - \mu_i)(X_j - \mu_i)^T \end{cases} \tag{9}$$

This differential equations are very hard to solve, so to combat this, we need to use a iterative process, the Expectation-Maximization (EM) algorithm,

The EM algorithm, first step define the initial conditions $\{\pi_i^{(0)}, \mu_i^{(0)}, \Sigma_i^{(0)}\}$, the second step is to calculate the posteriori probability, in the $l$-th iteration, $q_i^{(l)}(X_j) = \frac{\pi_i^{(l)} p_i^{(l)}(X_j)}{\sum_{k=1}^{K} \pi_k^{(l)} p_k^{(l)}(X)}$, the third step is to recalculate the quantities, $\{\pi_i^{(l+1)}, \mu_i^{(l+1)}, \Sigma_i^{(l+1)}\}$, and the fourth step is to stop when the values start converging.

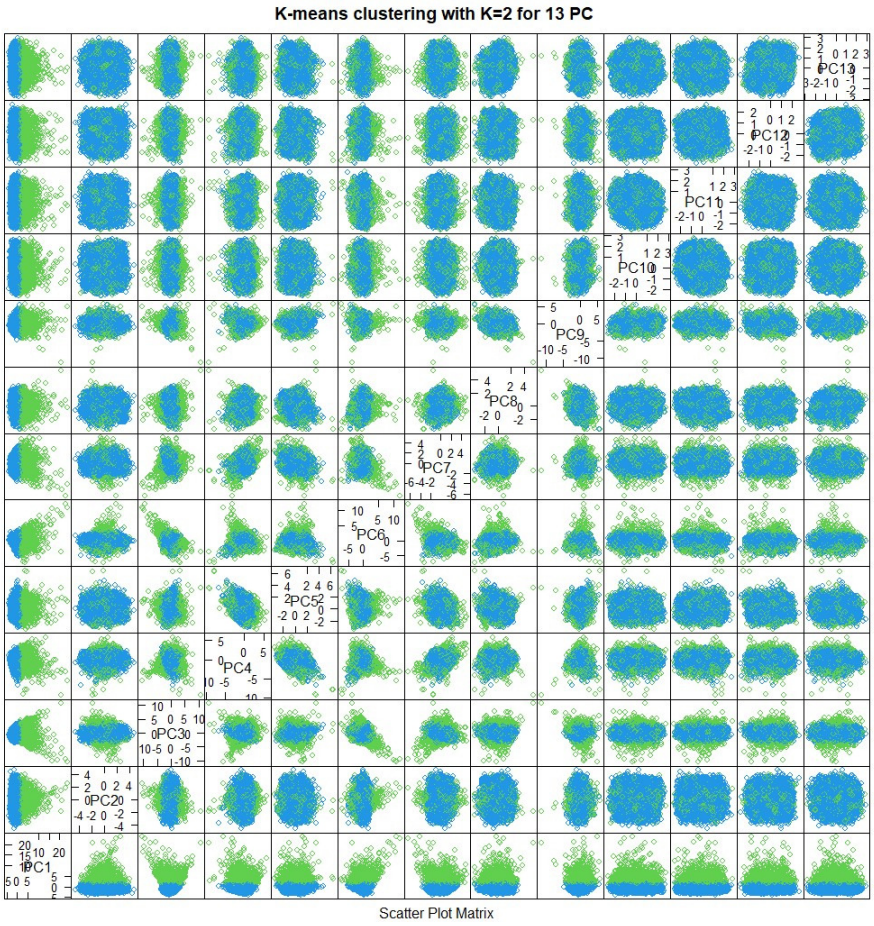



Figure 11: Graphical representation of the K-means clustering for every combination of principal components.

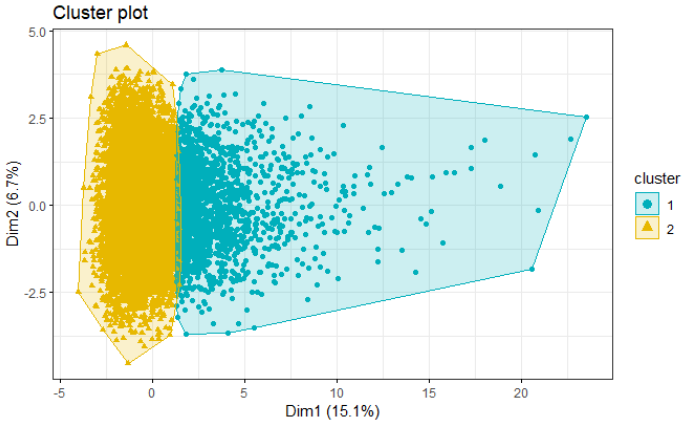



Figure 12: Cluster plot for the first and second principal components.

### 7.1 Analysis

In this section the built-in function in R to be used will be **mclust**

## REFERENCES

A. Asensio Ramos, M.C.M. Cheung, I. Chifu, and R. Gafeira. Machine learning in solar physics. *Living Reviews in Solar Physics*, 20:4, July 2023. doi:https://doi.org/10.1007/s41116-023-00038-x.

P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5:4308, July 2014. doi:https://doi.org/10.1038/ncomms5308.

Trevor Hastie. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 01 2009. ISBN 9780387848570. doi:10.1007/978-0-387-84858-7.

Alboukadel Kassambara. *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*, volume 1. STHDA, 2017.

Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(2):411–423, 2001. ISSN 13697412, 14679868. URL `http://www.jstor.org/stable/2680607`.

Daniel Whiteson. HIGGS. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5V312.