Pages 1–13
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 project was the application of statistical methods to a large-scale binary classification data set for Higgs boson detection. We started by finding the ideal number of principle components and examining the efficacy of Principle Component Analysis (PCA) as a dimensionality reduction method, using three commonly used criteria. Then, several clustering techniques were used to find patterns in the data, being them K-means clustering, hierarchical clustering and normal (Gaussian) mixture clustering. For these clustering approaches, we considered the principal components kept by the PCA, instead of using the original data, and we applied some cluster validity indices to find the optimal number of clusters. We explored some resulting statistics and some plots for each of the clustering techniques.

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 detect it) 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 lead to the creation of:

- 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.

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.

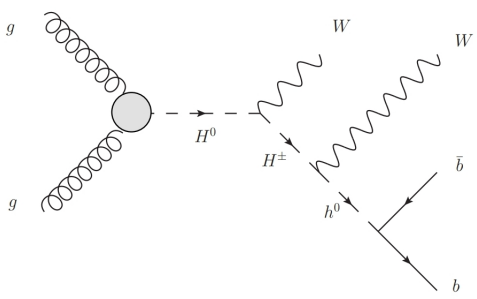

Figure 1: Signal Process. Extracted from Baldi et al. [2014].

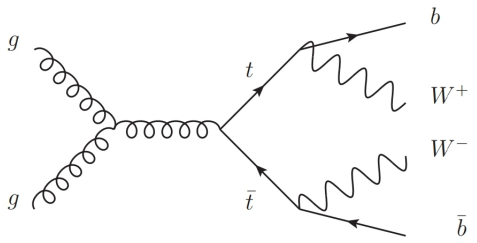

Figure 2: Background Process. Extracted from Baldi et al. [2014].

1.3 Input variables

The final product's properties that we can detect are the transverse momentum, p_T, and 2 angles ϕ, θ for the direction, where θ is rewritten as the pseudo-rapidity, $\eta = \tan(\theta/2)$. This way, we have 4 variables (p_T, ϕ, η, b-tag) for each jet and 3 variables (p_T, ϕ, η) for the lepton. We exclude the neutrino, of which we can only detect indirectly through the missing momentum and the angle ϕ. 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 (Baldi et al. [2014]).

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. It can be found in the UCI Machine Learning Repository at `archive.ics.uci.edu/ml/datasets/HIGGS` (Baldi et al. [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 command **summary** in R. For the target column, whether the process results in a Higgs particle (1) or 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** (R Core Team [2013]), to compute the principal components. It is also very important to note our data needed to be scaled, as in the Table 1 we see different variables have very different ranges.

The correlation matrix of the original variables in the data is given by figure 3. 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.

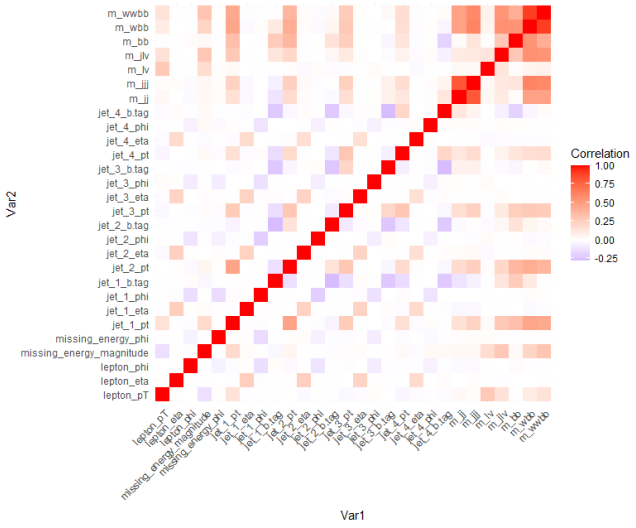

Figure 3: Correlation matrix of original variables using Pearson correlation coefficient.

And we can see that the original variables with the highest eigenvalue and most correlated (Figure 3) are the ones that most influence the PCs as seen in Figure 4.

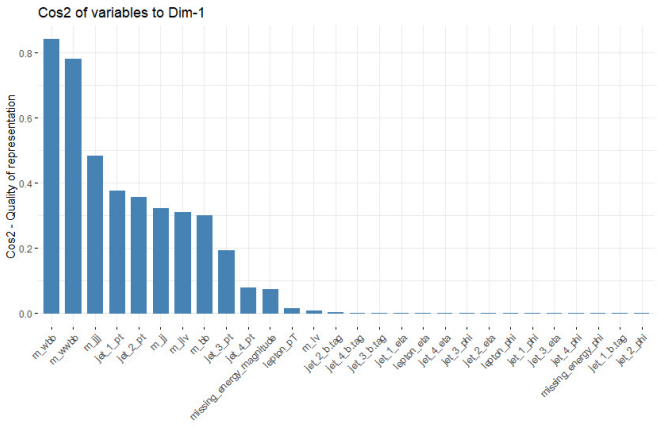

Figure 4: Dependence of the First PC with the original variables

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 5.

As the principal components are linear combinations of the original variables, with different weights, it is important to keep

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

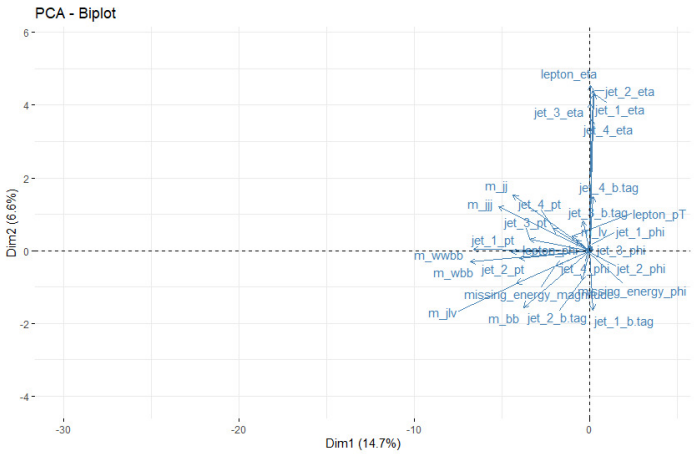

Figure 5: Biplot with the correlation between variables

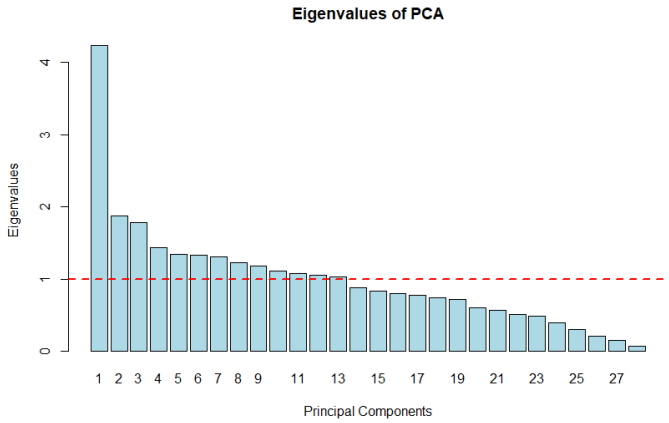

Figure 6: Eigenvalues of the principal components, with the horizontal line illustrating the cutoff of Kaiser-Guttman criterion.

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 6 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 7 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 need to have a small difference with respect to previous component, such that $\lambda_\alpha - \lambda_{\alpha-1} < \epsilon$. Looking at figure 6 again, there is no possible ϵ 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 8 shows a parallel coordinates plot for the 13 principal components that were kept with this criterion.

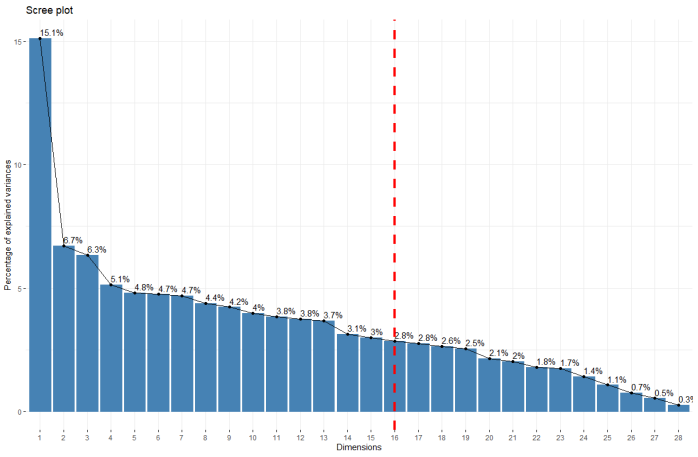

Figure 7: Percentage of variance explained by the principal components, with the vertical line illustrating the cutoff of the Pearson criterion.

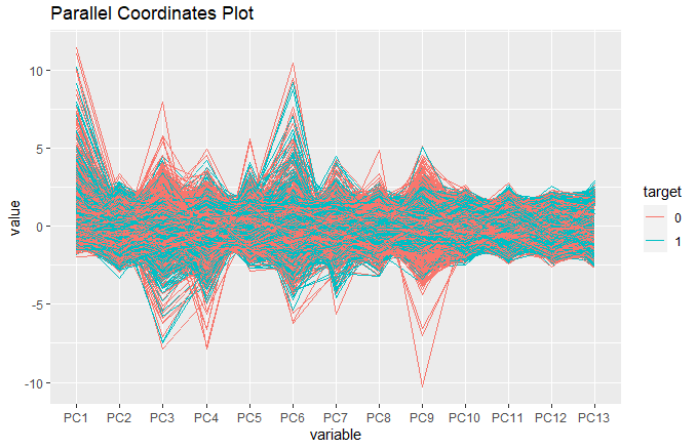

Figure 8: Parallel coordinates plot for the 13 principal components that were kept, colored according to if the event produces a Higgs or not.

4 CLUSTERING

Clustering is a technique in machine learning that looks for the similarity in data points and sets them 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 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 $d(x_i, x_j)$ between each pair of observations, one searches for the specific encoder $C(i)$ that accomplishes the necessary goal. 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. 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 (MacKay [2003]):

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, such as: 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]), in this case the general objective is to minimize the loss function, defined in Equation 2

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

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

- **Elbow method**: 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 9 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 10 shows us the average silhouette width with respect to the number of clusters. It shows us the optimal number of clusters with this method is $K = 2$ clusters (Rousseeuw [1987]).
- **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 11 allows us to see the optimal number of clusters using this test is $K = 1$ cluster.

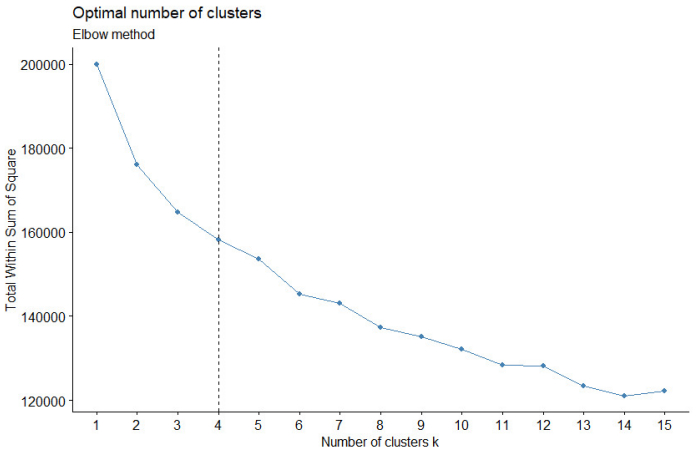

Figure 9: Elbow method: Total loss function in terms of the number of clusters

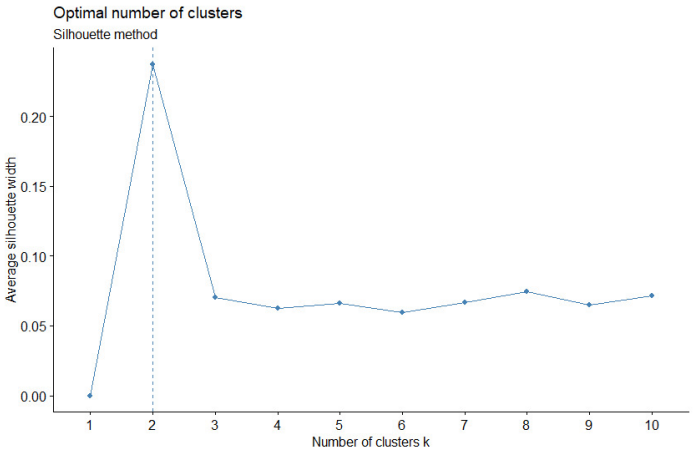

Figure 10: Silhouette method: Average silhouette width in terms of the number of clusters

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 $K = 2$.

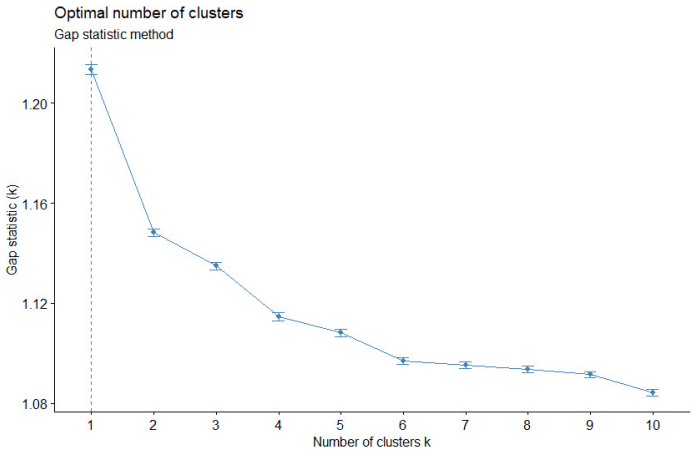

Figure 11: Gap statistic: Variation of the total loss function with the expected values of the data in terms of the number of clusters

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 $K = 2$.

5.3 Analysis

We used the R built-in-function **kmeans** (R Core Team [2013]) 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 12 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.

The within-cluster sum of squares is a measure of the compactness of clusters, providing information about how well the data points within each cluster are clustered together. Smaller values indicate that data points within each cluster are close to their respective centroids, which is desirable in k-means clustering. The result was 56 265.79 for the first cluster and 119 201.10 for the second, very high values.

In it's turn, the between-cluster sum of squares measures the separation between clusters and a larger value indicates that the clusters are distinct from each other. The ratio of between-cluster sum of squares with total sum of squares (between SS / total SS) is used as a measure of the quality of the clustering. It indicates how much of the total variance in the data is explained by the clustering, being 11.9%.

We can also observe that the number of points assigned to each cluster is severely unbalanced: while the original points are nearly 50%/50% with respect to the target label, the first cluster has around 1600 points and the second around 8300. We can also see in table 2 that there is no relation between the label (0 or 1) of a point and the cluster it was assigned to.

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

Table 2: Label of the original points and the cluster they were assigned to.

	Target	
Cluster	0	1
1	1034	898
2	3653	4415

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 dissmilarity (Hastie [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 inter-group 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 Analysis

We used the R built-in-function, **hclust** (R Core Team [2013]), that uses a agglomerative approach. It's important to note that we can use a variety of methods, and distance measures, but we have to be mindful of the data set and previous tools used. In another note as the methods use prototypical tree, the methods will consistently differ from each other and therefore comparing them is redundant. So, we used the same distances as mentioned in Section 4.1: Euclidean, Manhattan and the Chebychev. For the methods, we used the Ward's2, complete and average methods. In figures 15, 14 and 16, we look at the dendrograms resulted from using the Ward's method, we show that the results are inconclusive but can be cutted down to $K = 2$ clusters.

In figures 18, 17 and 19, we look at the dendrograms resulted from using the complete method, we show that the results have no apparent structure.

In figures 21, 20 and 22, we also look upon the dendrograms resulted from using the average method, and show that the results have no apparent structure, as well as the complete.

We also visualized heatmaps with the same combinations as for the dendrograms. As we they were all very similar and they didn't yield any interesting result, we just show one of the examples in figure 23. For comparison, figure 24 shows a heatmap of the original features with the same combination of method and distance.

7 NORMAL MIXTURE MODELS FOR CLUSTERING

The Gaussian (or normal) mixture model (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,

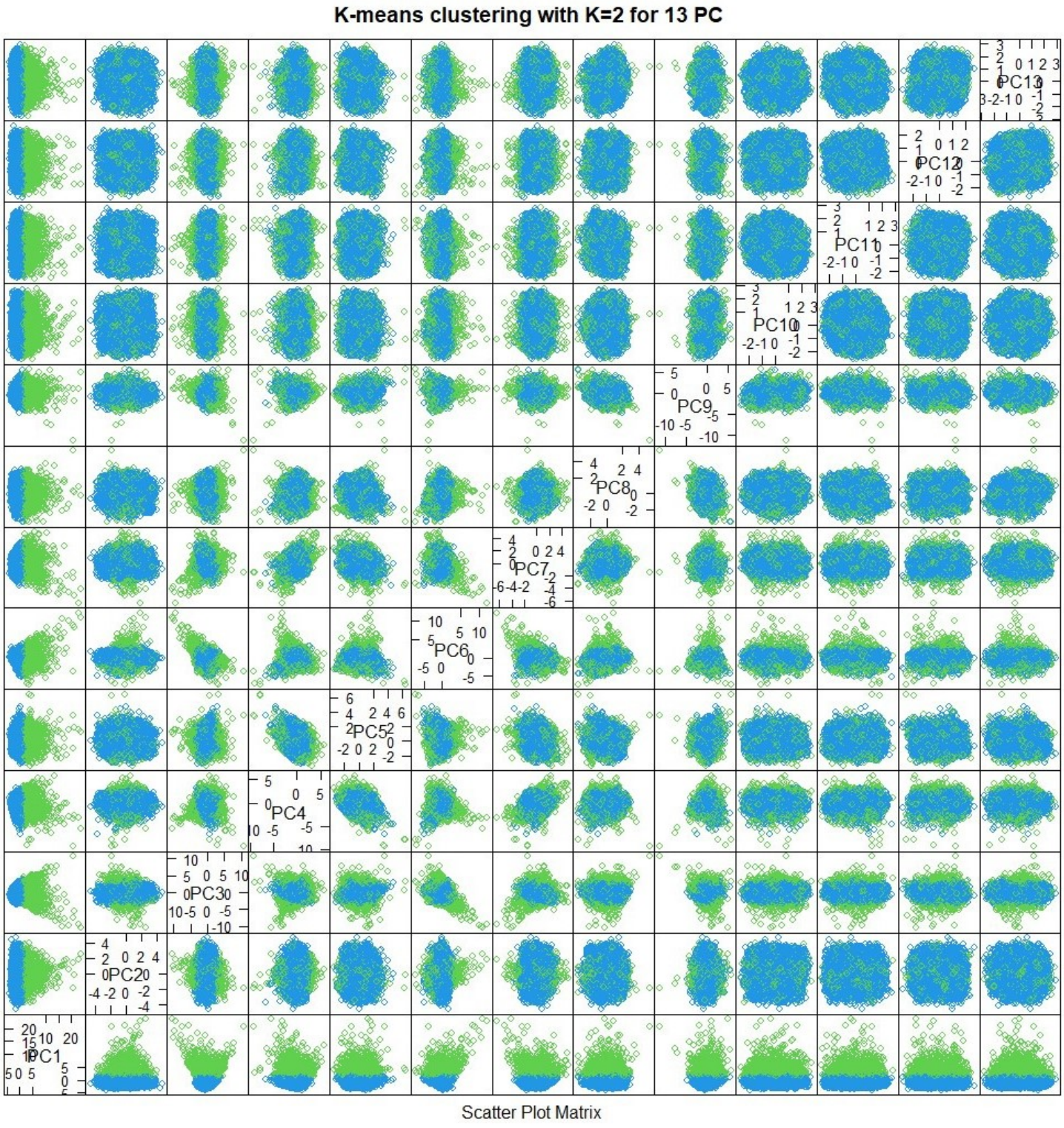

Figure 12: Graphical representation of the K-means clustering for every combination of principal components.

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 π_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.

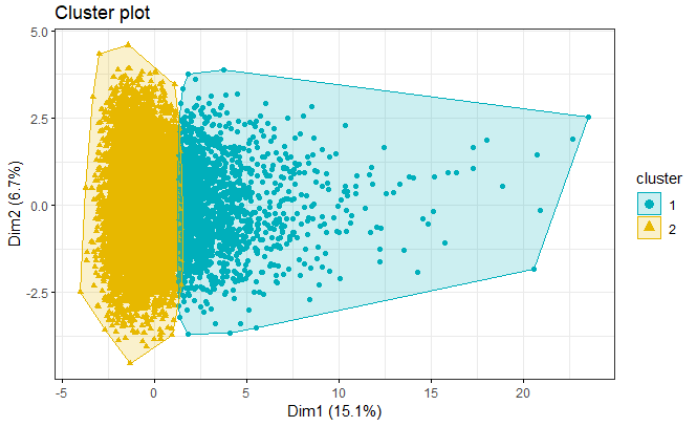

Figure 13: Cluster plot for the first and second principal components.

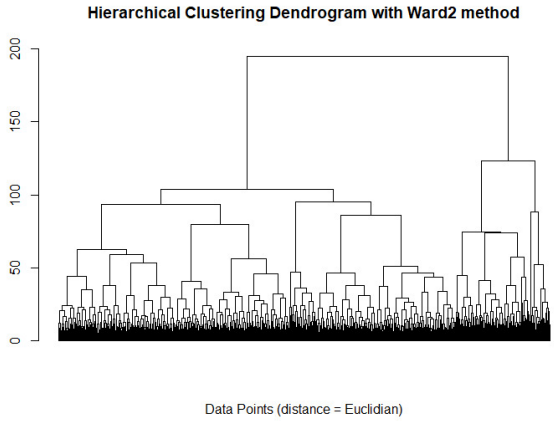

Figure 14: Dendrogram: Ward's method with the Euclidean distance.

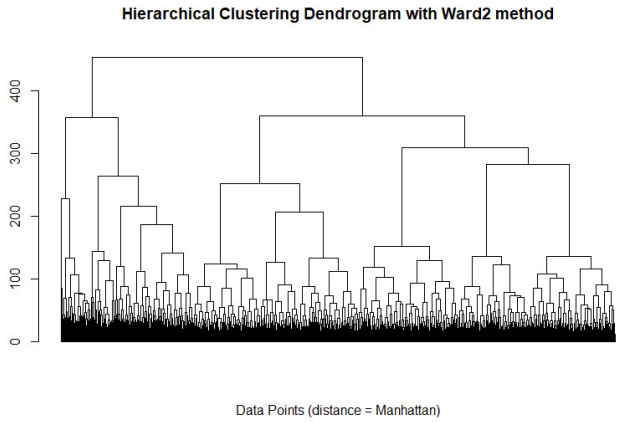

Figure 15: Dendrogram: Ward's2 method with the Manhattan distance.

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

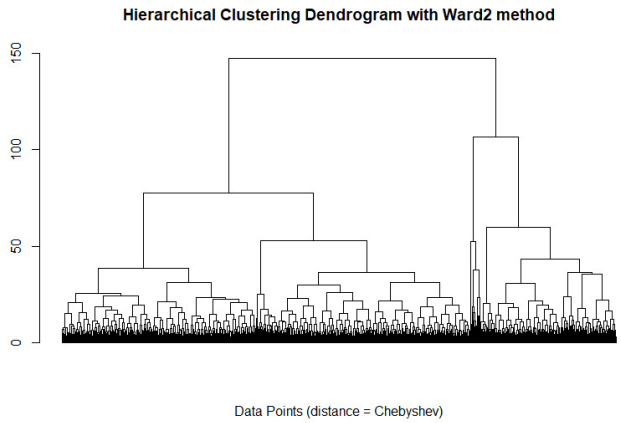

Figure 16: Dendrogram: Ward's method with the Chebychev distance.

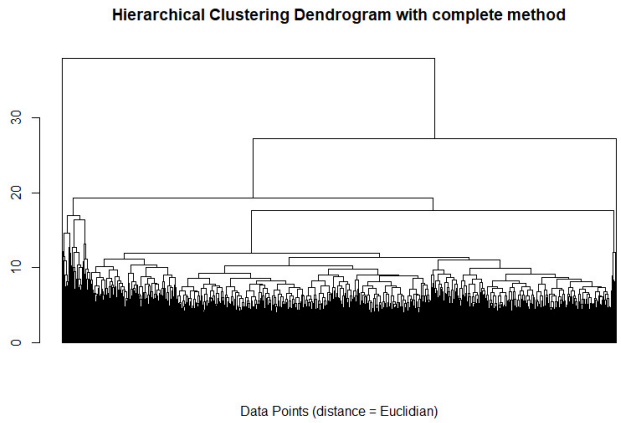

Figure 17: Dendrogram: Complete method with the Euclidean distance.

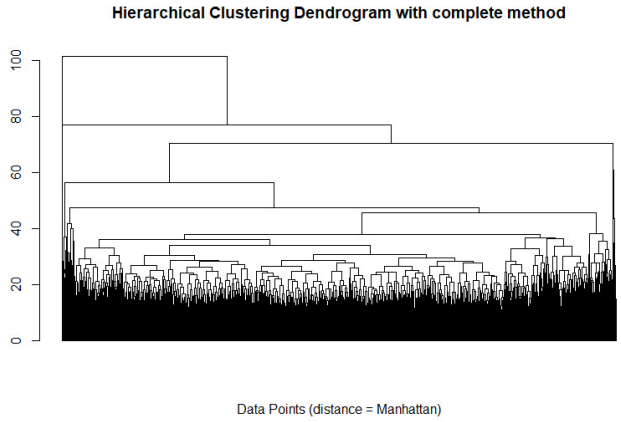

Figure 18: Dendrogram: Complete method with the Manhattan distance.

by,

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

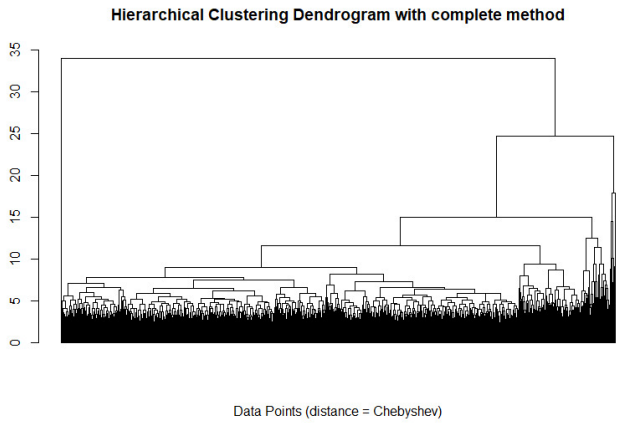

Figure 19: Dendrogram: Complete method with the Chebychev distance.

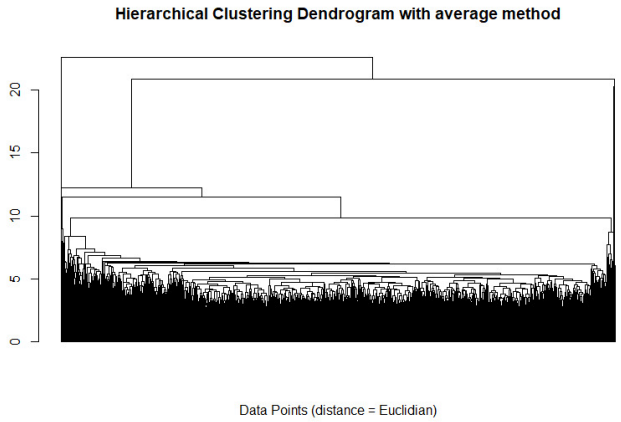

Figure 20: Dendrogram: Average method with the Euclidean distance

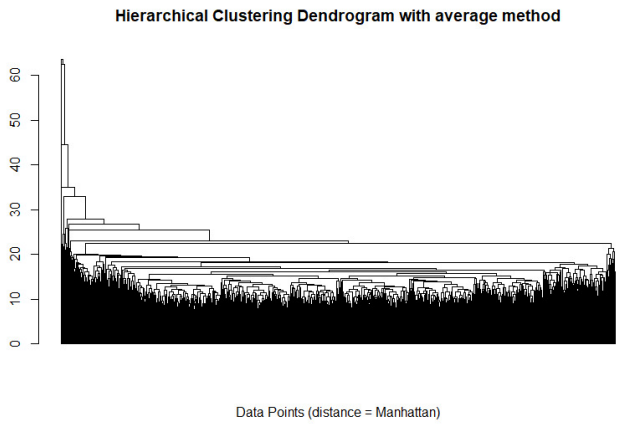

Figure 21: Dendrogram: Average method with the Manhattan distance.

In order to maximize this method, we use the Lagrange multiplier (Biernacki et al. [2003]),

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

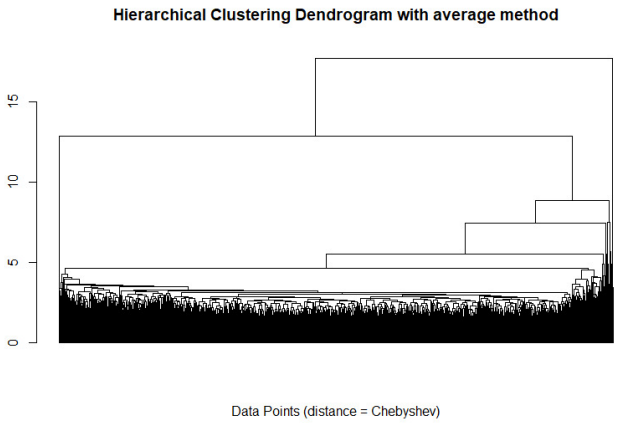

Figure 22: Dendrogram: Average method with the Chebychev distance

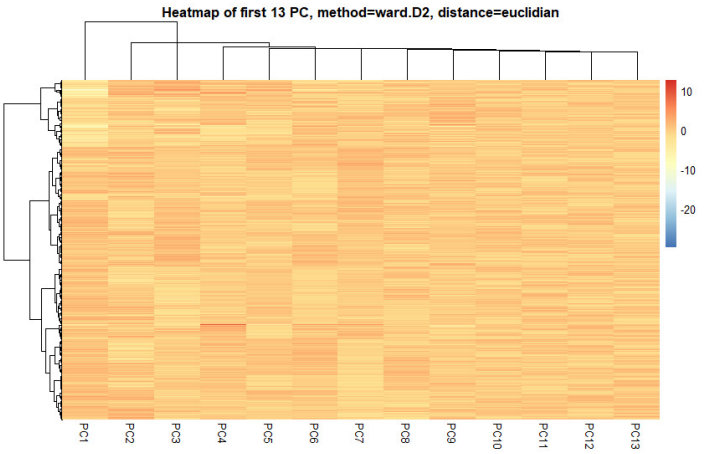

Figure 23: Heatmap: Ward D2 method with the Euclidean distance for first principal components.

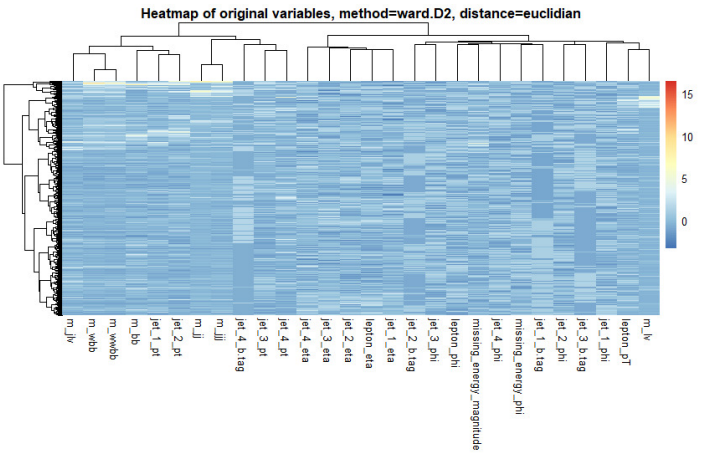

Figure 24: Heatmap: Ward D2 method with the Euclidean distance for the original variables.

where $p(X_j) = \sum_{i=1}^{K} \pi_i p_i(X_j)$, and λ 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

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

This system of equations is very hard to solve, so to combat this, we need to use a iterative process, the Expectation-Maximization (EM) algorithm (Biernacki et al. [2003]).

The EM algorithm, first step define the initial conditions $\{\pi_i^{(0)}, \mu_i^{(0)}, \Sigma_i^{(0)}\}$, the second step (or called E-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 (or called the M-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. Notice that with each iteration the log-likelihood of our model,

$$\log(p(X)) = \sum_j \log\left(\sum_{i=1}^{K} \pi_i N_{x_j}(\mu_i, \Sigma_i)\right) \tag{9}$$

which is guaranteed to converge.

7.1 Analysis

In this section, the built-in function in R used was **mclust** (Scrucca et al. [2016]), which is a mix of the Maximum Likelihood estimator and the Expectation-Maximization algorithm. The algorithm provides various functions to estimate the parameters $\{\pi_i, \mu_i \Sigma_i\}$. The EM algorithm proceeds to search over a range of different models in a multivariate mixture where for example: "EII" (spherical, equal volume), "VEE" (ellipsoidal, equal shape and orientation), and more, were used. And in order to evaluate how fitting these models are to the data we use the Bayesian Information Criterion (BIC).

In Figure 25, we show how well the various models fit to the data, being the one that represents the best fit the "VVV" (ellipsoidal, varying volume, shape and orientation), when the number of components is 10. Figure 26 shows the implementation of this best fit model for normal mixture clustering for every combination of principal component. In table 3, we can see some resulting statistics of the GMM model obtained: the log-likelihood, the BIC criterion, the number of estimated parameters df and the ICL (Integrated Complete-data Likelihood, which is essentially the ordinary BIC penalized by the subtraction of the estimated mean entropy). For BIC and ICL, the higher the better the model is fitted. In the same table we can also see the number of points attributed to each cluster and the correspondent mixing probabilities.

For comparison, we performed the same modelling but defining that the BIC is only calculated for the first two mixture components, to see if there's any visual connection with the binary classification problem. Figure 27 shows the various model fit, being the best one the "VVE" (ellipsoidal, equal orientation), for two components. Figure 29 shows the implementation of this model for normal mixture clustering. In table 4, we can see some resulting statistics of the GMM model obtained, like previously. Figure 29 shows a classification for these two principal components.

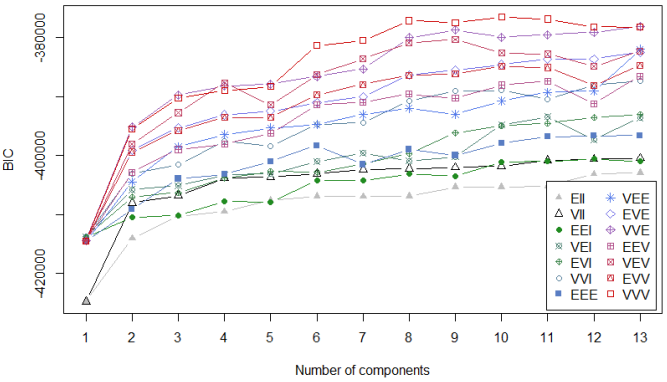

Figure 25: Bayesian Information Estimator (BIC) as a function of the PCs for various models

8 CONCLUSION

The objective of this project was the application of statistical methods to a large-scale binary classification data set for Higgs boson detection, with 28 physical features.

We started by finding the ideal number of principle components and examining the efficacy of Principle Component Analysis (PCA) as a dimensionality reduction method. The validity of three commonly used criteria (Kaiser-Guttman, Pearson and Cattell's) were evaluated and the Kaiser-Guttman criterion was chosen, keeping 13 principal components.

Then, several clustering techniques were to try to find patterns in the data. The first approach discussed was K-means clustering. To find the optimal number of clusters, we used the Elbow method, the Silhouette method and the gap statistic. We opted for the Silhouette method, that returned K=2. According to the preliminary clustering results, a distinct, non-overlapping cluster could be seen when comparing the first PC with the other twelve. Nevertheless, when the remaining PC combinations are examined, the clusters often overlap.

The second technique was hierarchical clustering, that doesn't require an initial value of clusters, but distance measures. For this, we considered 3 linkages (complete, average and Ward's methods) and 3 distance metrics (Manhattan, Euclidean and Chebychev) and, according to the Ward's method, $K = 2$. For any of the combinations, there wasn't any visible apparent structure.

The third technique explored was normal (Gaussian) mixture clustering for the 13 PC kept, that uses the maximum likelihood estimator and the Expectation-maximization algorithm to estimate the Gaussian parameters. With this, we analysed some statistics, like the BIC, and found that the best fit model was "VVV". We repeated the same study for only the first two PC, for comparison.

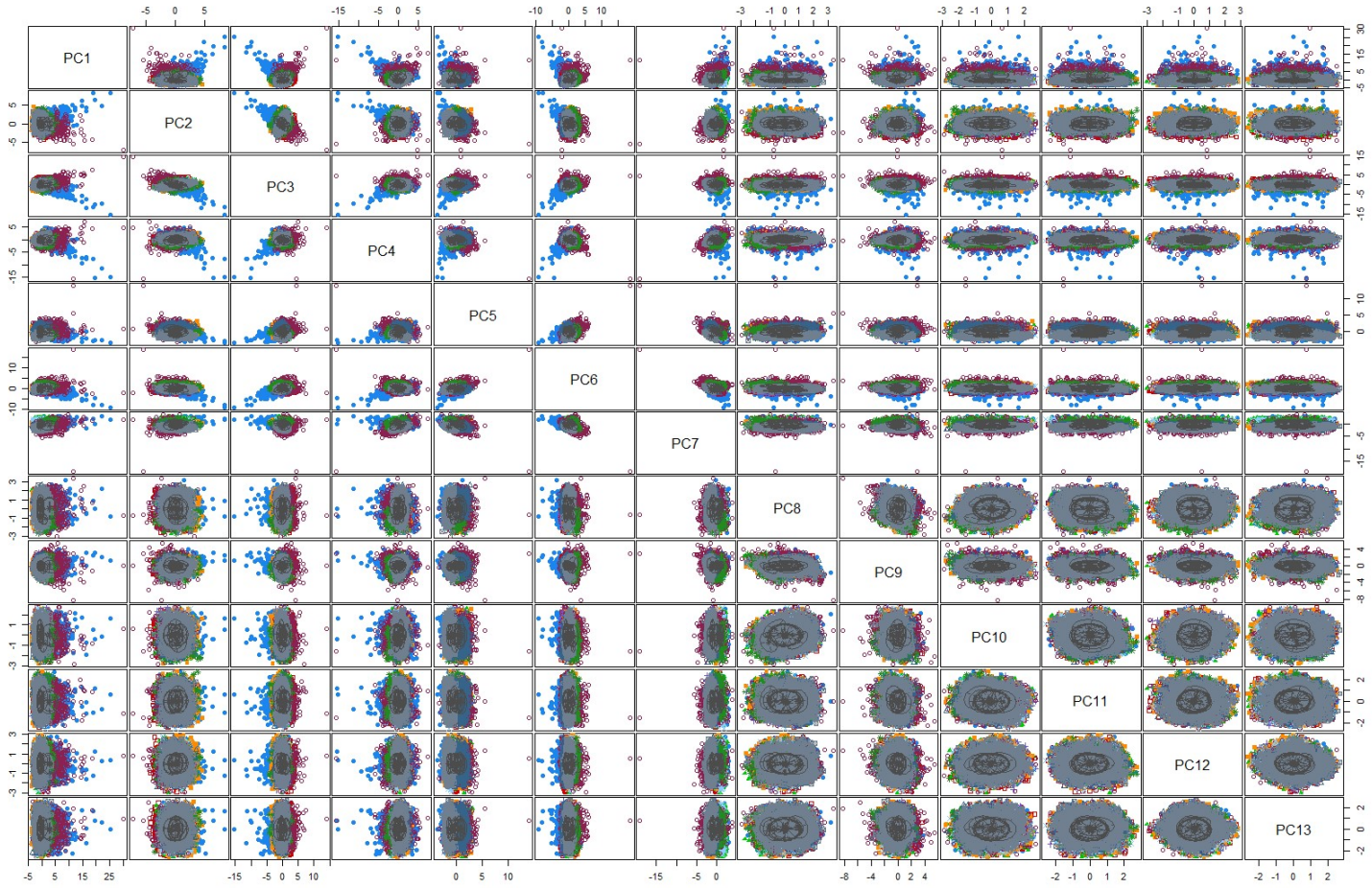

Figure 26: Graphical representation of the normal mixture clustering using the ”VVV” model between every combination of PCs.

Table 3: Statistics for Gaussian mixture model fitted by EM algorithm, mixing probabilities and number of points for each cluster.

log-likelihood		**n**		**df**		**BIC**		**ICL**	
-183450.1		10000		1049		-376561.9		-378118.6	
Clustering table:									
1	2	3	4	5	6	7	8	9	10
1003	1543	1578	631	1665	1167	638	239	651	885
Mixing probabilities:									
1	2	3	4	5	6	7	8	9	10
0.10349	0.15274	0.15542	0.06263	0.16226	0.11354	0.07367	0.02561	0.06364	0.08697

Table 4: Statistics for Gaussian mixture model fitted by EM algorithm, mixing probabilities and number of points for each cluster.

log-likelihood	**n**	**df**	**BIC**	**ICL**
-196996.3	10000	131	-395199.2	-396060
Clustering table:				
1			2	
8267			1733	
Mixing probabilities:				
1	2			
0.8154968	0.1845032			

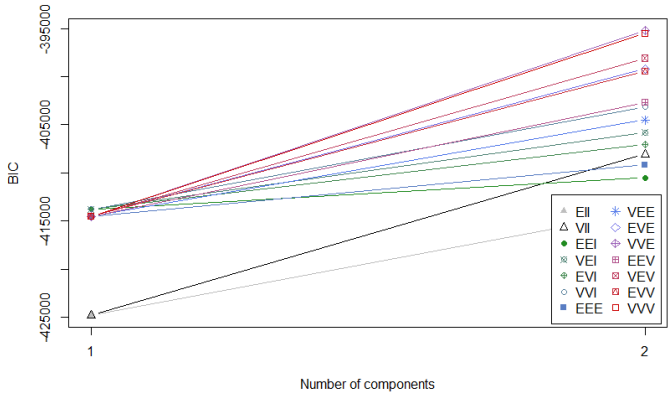

Figure 27: Bayesian Information Estimator (BIC) as a function of the PCs for various models for the first two principal components.

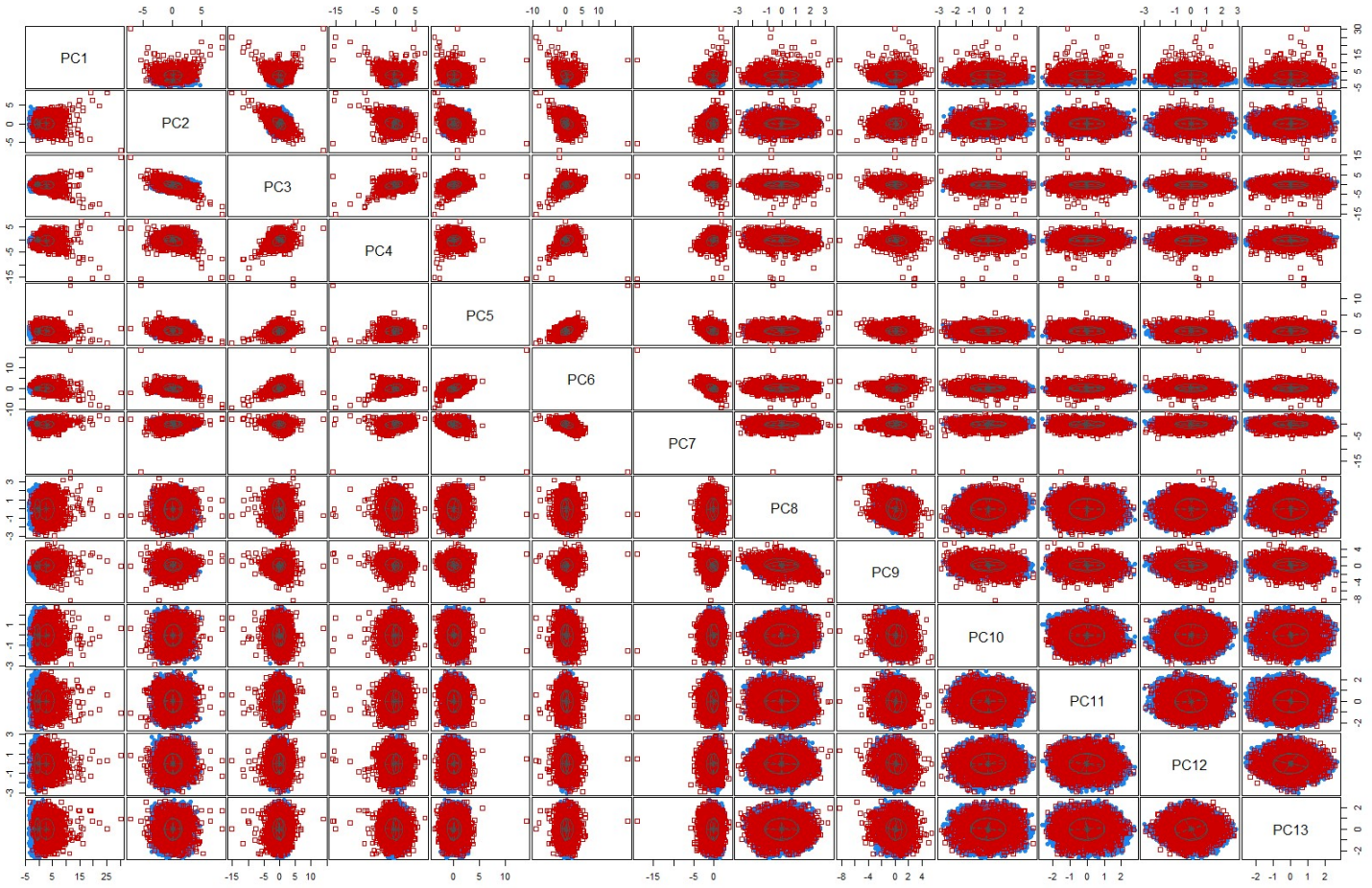

Figure 28: Graphical representation of the normal mixture clustering using the ”VVE” model between every combination of PCs.

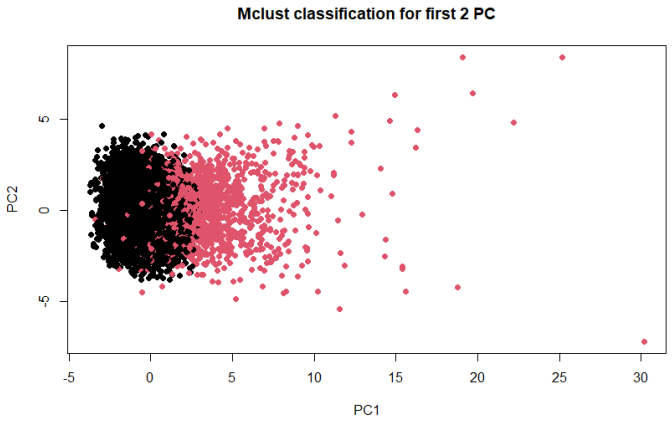

Figure 29: Mclust classification for the first two principal components.

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.

Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics Data Analysis*, 41(3):561–575, 2003. ISSN 0167-9473. doi:https://doi.org/10.1016/S0167-9473(02)00163-9. URL https://www.sciencedirect.com/science/article/pii/S0167947302001639. Recent Developments in Mixture Model.

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.

David MacKay. *InformationTheory, Inference, and Learning Algorithms*, volume 50. 01 2003. ISBN 978-0-521-64298-9. doi:10.1109/TIT.2004.834752.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL http://www.R-project.org/. ISBN 3-900051-07-0.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi:https://doi.org/10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257.

Luca Scrucca, Michael Fop, T. Brendan Murphy, and Adrian E. Raftery. mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1):289–317, 2016. URL https://doi.org/10.32614/RJ-2016-021.

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.