Pages 1–7
2023

Project 2 - 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 supervisioned classification methods to a large-scale binary classification data set for Higgs boson detection. We started by applying linear and quadratic discriminant analysis (LDA and QDA), finding accuracies of 64.5% and 64.3%, respectively. Next, we applied a decision tree, obtaining a maximum accuracy of 68.8%. The third method used was the k-nearest neighboors (kNN), with a maximum accuracy of 63.7%. Finally, we used a support vector machine (SVM) with different kernels, obtaining an accuracy of 68.5%. We explored some resulting statistics and some plots for each of technique.

CONTENTS

1 INTRODUCTION

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 applied different supervisioned classification techniques using R (R Core Team [2013]). We adapted the number of events (data set rows) in order to use the maximum computationally feasible number, often being 50 000 randomly chosen events. Figure 1 consists in a histogram showing the distribution of the numeric variables of 50 000 randomly chosen events from the original data set.

We noticed that the variables *jet_1_b.tag*, *jet_2_b.tag*, *jet_3_b.tag* and *jet_4_b.tag* were used in the Project 1 as numeric variables, but this is not the case. These tags correspond to categorical variables with respect to the jets in the physics process. This error was corrected in this work by converting the variables to categorical.

2 LINEAR AND QUADRATIC DISCRIMINANT ANALYSIS

Linear and quadratic discriminant analysis, LDA and QDA, respectively, are a type of classification problem, based on linear decision boundaries. In order to know the optimal classification we need to look at the class G **posteriori** probability, for a given point, X, $P(G|X)$, by Bayesian inference, the Bayesian theorem says that

$$P(G = k|X = x) = \frac{P(X = x|G = k)P(G = k)}{P(X = x)} \quad (1)$$

$$= \frac{f_k(x)\pi_k}{\sum_{l=1}^{k} \pi_k f_k(x)} \quad (2)$$

where π_k is the **priori** probability, which is the proportion of points with k class, $\sum_{k=1}^{K} = 1$, and $f_k(x)$ is assumed to be a normal distribution,

$$f_k(x) = \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)} \quad (3)$$

where the covariance matrix Σ_k, in LDA is $\Sigma_k \equiv \Sigma$, hence the same for every class, but we can't generalize this to QDA. Our objective would be to maximize the posteriori probability, and using the maximum likelihood estimator, we will maximize this function

$$\sum_k \sum_j \log p_k(x_{kj}) + \sum_k N_k \log \pi_k - \lambda \left(\sum_k \pi_k - 1 \right) \quad (4)$$

2.1 LDA

In order to perform a LDA, we used the built-in function of R **lda** (Brian Ripley [2023]), and divided our data into 70% train and 30% test. In Table 1, we show the confusion matrix.

In Table 2, the accuracy (i.e. the sum of the diagonal of the confusion matrix divided by the total number of points), the

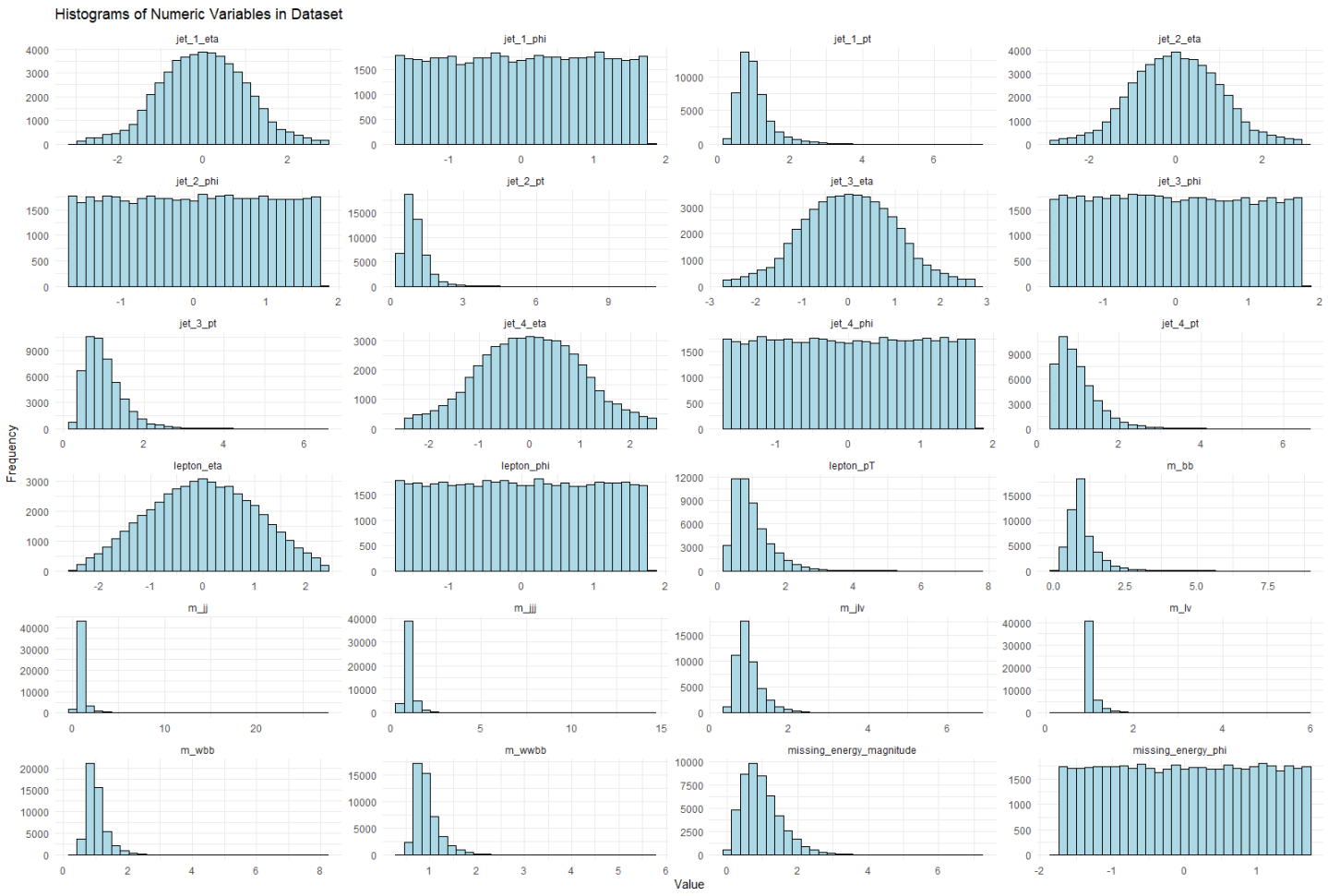

Figure 1: Histogram of numerical variables of 50 000 randomly chosen events from the original data set.

Table 1: Confusion matrix for LDA.

	Reference	
Prediction	0	1
0	2410	1257
1	2292	4041

Table 2: Statistics for LDA: accuracy, sensitivity, specificity and Cohen's kappa coefficient.

Accuracy	**Sensitivity**	**Specificity**	κ
0.6451	0.5125	0.7627	0.2787

sensitivity (true positive rate), the specificity (true negative rate) and the Cohen's kappa coefficient κ. This last one is a statistic that is used to measure inter-rater reliability (and also intra-rater reliability) for qualitative (categorical) items (ML [2012]). It is generally thought to be a more robust measure than simple percent agreement calculation, as κ takes into account the possibility of the agreement occurring by chance (Pontius and Millones [2011]). In this case, the traditional 2×2 confusion matrix to evaluate binary classifications, the Cohen's Kappa formula can be written as:

$$\kappa = \frac{2 \times (\mathrm{TP} \times \mathrm{TN} - \mathrm{FN} \times \mathrm{FP})}{(\mathrm{TP} + \mathrm{FP}) \times (\mathrm{FP} + \mathrm{TN}) + (\mathrm{TP} + \mathrm{FN}) \times (\mathrm{FN} + \mathrm{TN})}, \quad (5)$$

where TP are the true positives, FP are the false positives, TN are the true negatives, and FN are the false negatives (Chicco et al. [2021]).

In Table 2, the low sensitivity and high specificity shows that LDA tends to correctly categorize when there is an Higgs produced (1) more than when the Higgs is not produced (0).

2.2 QDA

In order to perform a QDA, we used the built-in function of R **qda**, and divided our data into 70% train and 30% test. In Table 3, we show the confusion matrix.

Table 3: Confusion matrix for QDA.

	Reference	
Prediction	0	1
0	2118	983
1	2584	4315

In Table 4, the accuracy, the sensitivity, the specificity and the Cohen's kappa coefficient κ. In Table 4, the low sensitivity and high specificity shows that QDA tends to correctly categorize when there is an Higgs produced (1) more than when the Higgs is not produced (0).

Table 4: Statistics for QDA: accuracy, sensitivity, specificity and Cohen's kappa coefficient.

Accuracy	Sensitivity	Specificity	κ
0.6433	0.4504	0.8145	0.2701

3 DECISION TREE

Tree-based methods partition the feature space into a set of rectangles, and then fit a simple model (like a constant) in each one. They are conceptually simple yet powerful, and a popular method for tree-based classification is called CART, applied in this work, with the R package *rpart* (Hastie [2009], Terry Therneau [2023]). The default splitting index used was the default, the Gini index, a measure of the node impurity:

$$i(t) = \sum_{i \neq j} p_i p_j = 1 - \sum_{j=1}^{K} p_j^2, \tag{6}$$

where $p_i = \text{Prob}(\omega_i|t)$, ω_i is the son node and t is the father node.

We considered a grid of possible parameters for the four parameters of the decision tree trained: *minsplit* (minimum number of observations that must exist in a node in order for a split to be attempted), *minbucket* (minimum number of observations in any terminal leaf node), *maxdepth* (maximum depth of any node of the final tree, with a maximum of 26 depth) and *cp* (complexity parameter, where any split that does not decrease the overall lack of fit by a factor of *cp* is not attempted) . The values are shown in table 5 and were chosen in order to cover a good amount of values, simultaneously preserving a doable computing time.

Table 5: Possible values for four parameters of the decision tree.

Parameter	Values
minsplit	1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 51
minbucket	1, 6, 11, 16, 21, 26,31
maxdepth	1, 6, 11, 16, 21, 26
cp	0.0005,0.001,0.005,0.01,0.05,0.1

We used 50 000 rows of the original data set and divided them into training and test data sets, 70% and 30%, respectively. First, we considered the "raw" un-scaled data. The confusion matrix obtained is shown in table 6. To have an idea of the influence of the tuning parameters in the accuracy obtained, figure 2 shows the accuracy as a function of the complexity parameter and the depth of the tree and 3 shows the accuracy as a function of the minimum split and the minimum bucket. For figure 2, if the depth is only 1 (two nodes), the accuracy doesn't improve no matter the complexity parameter. For the scenario where the data is scaled, there is almost no difference in these plots, so they aren't shown. The confusion matrix obtained is shown in table 7.

Figure 4 shows the importance of the variables for the model of the tree for un-scaled data and as one can see it is very similar to when the PCA was applied in the first project. Figure 5 shows the importance of the variables for the scaled data. The difference between the two scenarios is very small.

Table 8 shows the optimal tuning parameters for the two scenarios considered (scaled or un-scaled data). Table 9 shows some important statistics, like the accuracy, the sensitivity, the specificity and the Cohen's kappa coefficient κ. The performance metrics don't get better by scaling the data.

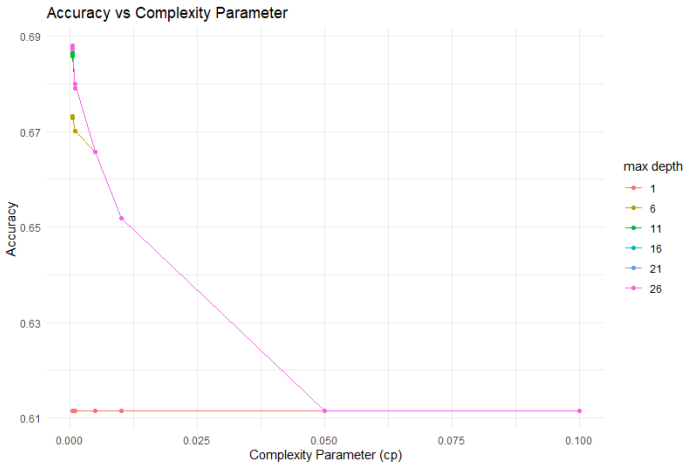

Figure 2: Accuracy as a function of complexity parameter, for different values of depth.

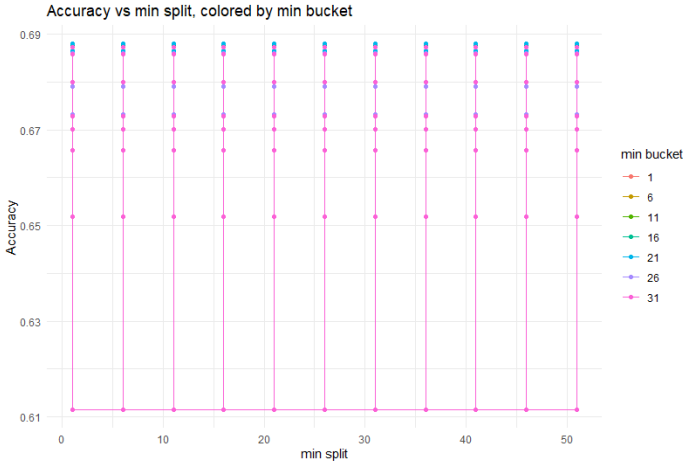

Figure 3: Accuracy as a function of minimum split, for different values of minimum bucket.

Table 6: Confusion matrix for un-scaled data for decision tree.

	Reference	
Prediction	0	1
0	4604	2251
1	2428	5716

Table 7: Confusion matrix for scaled data for decision tree.

	Reference	
Prediction	0	1
0	4596	2234
1	2490	5679

4 K-NEAREST NEIGHBOORS (KNN)

Nearest-neighbor methods use those observations in a training set closest in input space to x to form $\hat{Y}$. Specifically, the k-nearest neighbor fit for $\hat{Y}$ is defined as follows:

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i, \tag{7}$$

where $N_k(x)$ is the neighborhood of x defined by the k closest points x_i in the training sample. Closeness implies a metric, which

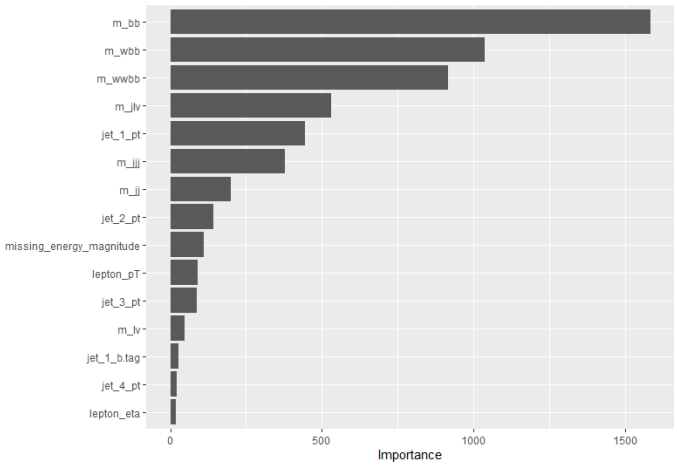

Figure 4: Importance of the first 15 variables of the un-scaled data set.

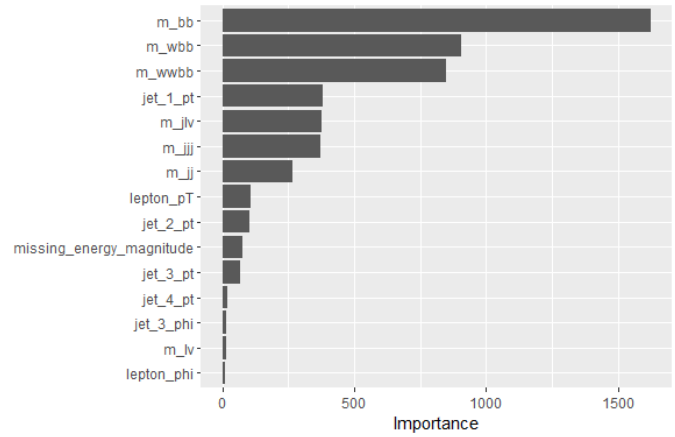

Figure 5: Importance of the first 15 variables of the scaled data set.

Table 8: Tuning parameters for both scenarios in decision tree (scaled or not): *cp*, *minsplit*, *minbucket* and *maxdepth*.

Scaled?	**cp**	**minsplit**	**minbucket**	**maxdepth**
No	0.0005	1	21	16
Yes	0.0005	1	31	11

Table 9: Statistics for both scenarios in decision tree (scaled or not): accuracy, sensitivity, specificity and Cohen's kappa coefficient.

Scaled?	**Accuracy**	**Sensitivity**	**Specificity**	κ
No	0.6880	0.6547	0.7175	0.3727
Yes	0.6850	0.6486	0.7177	0.3670

consists in the Euclidean distance. So, in other words, we find the k observations with x_i closest to x in input space, and average their responses (Hastie [2009]).

For the data set used in this project, we ran kNN models using the *caret* R library (Max Kuhn [2023]) for 50 000 rows of the original data set. We divided this sub data set in two: 70% for training and 30% for test. We also considered two scenarios: the first where we used the "raw" data, with no scaling, and the second one where we pre-processed the data by centering it with range bounds of 0 and 1.

To retrieve the best performing number of k neighbors, we ran a 10-fold cross-validation check, with k varying from 1 to 60. The accuracies obtained for the training data are shown in figures 7 and 8, for un-scaled and scaled training data, respectively. As we see, the accuracy reaches a plateau from around 30 k on-wards, being the optimal value of k was 47 for un-scaled data and 53 for scaled data. The confusion matrices for the test data in both scenarios are shown in tables 10 and 11, respectively.

Table 12 shows some important statistics, like the accuracy, the sensitivity, the specificity and the Cohen's kappa coefficient κ.

Looking at table 12, the accuracy increases by $\sim 4\%$ and the sensitivity by $\sim 9\%$ when we scale the data. The specificity, on the other hand, decreases by $\sim 2\%$. The high specificity and the low sensitivity indicate that the kNN model tends to correctly categorize when there is an Higgs produced (1) more than when the Higgs is not produced (0).

Table 10: Confusion matrix for un-scaled data for kNN.

	Reference	
Prediction	0	1
0	2285	1298
1	4731	6685

Table 11: Confusion matrix for scaled data for kNN.

	Reference	
Prediction	0	1
0	3109	1474
1	3970	6446

Table 12: Statistics for both scenarios in kNN (scaled or not): accuracy, sensitivity, specificity and Cohen's kappa coefficient.

Scaled?	**Accuracy**	**Sensitivity**	**Specificity**	κ
No	0.5980	0.3257	0.8374	0.1681
Yes	0.6370	0.4391	0.8139	0.2579

5 SUPPORT VECTOR MACHINES (SVM)

Support vector machine (SVM) is a supervised algorithm, this algorithm constructs a set of hyperplanes, such that they separate and classify the data. The main objective of SVM's is to computationally solve the kernel method, the main benefit is that it work with the non-linearity of the separable data, because it works with a multiple set of kernel functions, in this project we will use the **radial, linear, sigmoid** and **laplace** kernels.

The kernel method estimator is

$$\hat{E}[y \mid x] = \frac{\sum_{i=1}^{n} y_i \mathcal{K}(x, x_i)}{\sum_{i=1}^{n} \mathcal{K}(x, x_i)} \tag{8}$$

The SVM is the classifier of the weighted kernel given by,

$$\hat{y}(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i \mathcal{K}(x, x_i)\right) \tag{9}$$

And our objective is to choose the coefficients, α_i, such that the performance is the best, with the most zero coefficients possible.

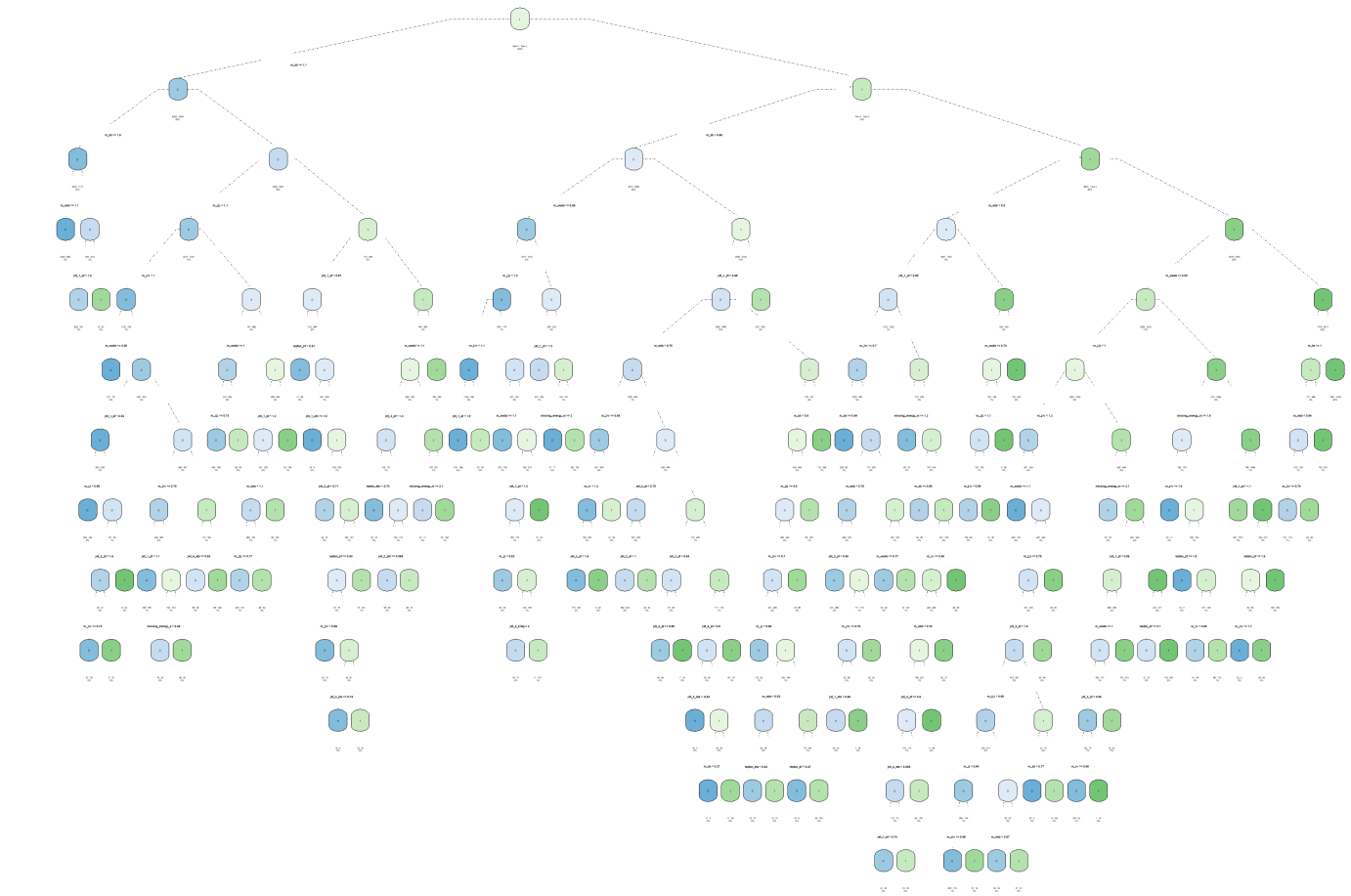

Figure 6: Scheme of decision tree for un-scaled data, where we can see the splitting rules, the nodes and leafs. Unfortunately, the tree is too deep to effectively visualize it.

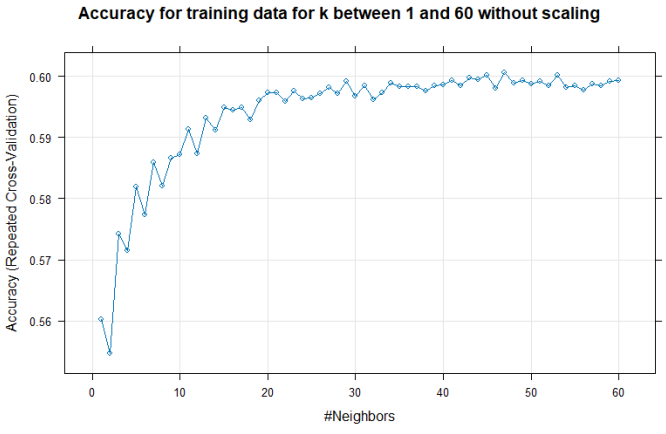

Figure 7: Accuracy for un-scaled training data using 10-fold cross-validation with k ranging from 1 to 60.

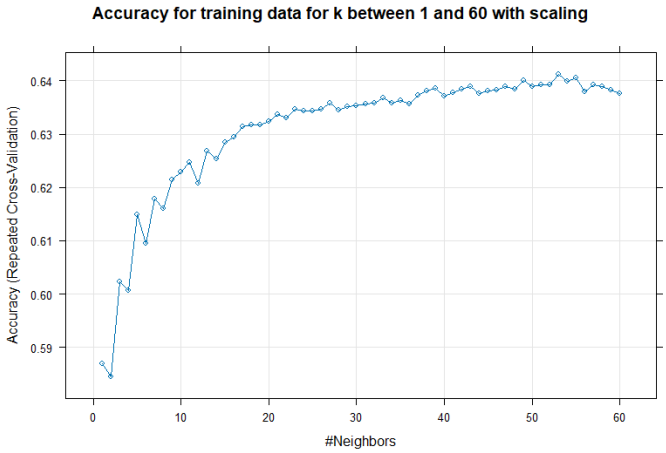

Figure 8: Accuracy for scaled training data using 10-fold cross-validation with k ranging from 1 to 60.

Using the built-in function in R, **svm** (David Meyer [2023]), and splitting our data into 70% train and 30% test. In Table 13, we show the confusion matrix with the radial kernel.

In Table 14, we show the confusion matrix with the linear kernel.

In Table 15, we show the confusion matrix with the sigmoid kernel.

Using the built-in function in R, **ksvm**, from the library, *kernlab* (Alexandros Karatzoglou [2023]). In Table 16, we show the confusion matrix with the laplace kernel.

In Table 17, the accuracy, the sensitivity, the specificity and the Cohen's kappa coefficient κ, for all the different SVM used. In Table 17, the low sensitivity and high specificity for the radial, linear and laplace kernels show that the SVM with those

Table 13: Confusion matrix for SVM with a radial kernel

	Reference	
Prediction	0	1
0	4145	1827
1	2905	6122

Table 14: Confusion matrix for SVM with a linear kernel

	Reference	
Prediction	0	1
0	3237	1516
1	3831	6416

Table 15: Confusion matrix for SVM with a sigmoid kernel

	Reference	
Prediction	0	1
0	3740	3311
1	3334	4614

Table 16: Confusion matrix for SVM with a laplace kernel

	Reference	
Prediction	0	1
0	3547	1513
1	3509	6431

respective kernel tends to correctly categorize when there is an Higgs produced (1) more than when the Higgs is not produced (0). While the SVM using the sigmoid kernel tends to give the poorest results.

Table 17: Statistics for the different kernels in SVM: accuracy, sensitivity, specificity and Cohen's kappa coefficient.

Kernel	**Accuracy**	**Sensitivity**	**Specificity**	κ
Radial	0.6845	0.5879	0.7702	0.3612
Linear	0.6435	0.4580	0.8089	0.2717
Sigmoid	0.5570	0.5287	0.5822	0.1109
Laplace	0.6652	0.5027	0.8095	0.3173

6 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 used different supervised techniques, tuning the hyper-parameters using methods like k-fold cross-validation or simple accuracy tuning.

We use the accuracy as the defining metric of the performance of the models, being the best versions of each technique applied shown in table 18. Every model except SVM with sigmoid kernel obtained an accuracy above 60%, being the best one the decision tree with un-scaled data, achieving an accuracy of 68.80%. Albeit not good enough to be confident, the results provide a glimpse of the limits of simple models in this specific data set, that depicts a simple answer (binary) - complex process situation (particle physics).

Table 18: Accuracies for the best models of each technique applied.

Model	**Scaled?**	**Accuracy (%)**
LDA	Yes	64.51
QDA	Yes	64.33
Decision Tree	No	68.80
kNN	Yes	63.70
SVM (Radial)	—	68.45

REFERENCES

Kurt Hornik National ICT Australia (NICTA) Michael A. Maniscalco Choon Hui Teo Alexandros Karatzoglou, Alex Smola. kernlab: Kernel-Based Machine Learning Lab. https://cran.r-project.org/package=kernlab, 2023. [Online; accessed 12-12-2023].

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.

Douglas M. Bates Kurt Hornik Albrecht Gebhardt David Firth Brian Ripley, Bill Venables. Support Functions and Datasets for Venables and Ripley's MASS. https://cran.r-project.org/package=MASS, 2023. [Online; accessed 12-12-2023].

Davide Chicco, Matthijs J. Warrens, and Giuseppe Jurman. The matthews correlation coefficient (mcc) is more informative than cohen's kappa and brier score in binary classification assessment. *IEEE Access*, 9:78368–78381, 2021. doi:10.1109/ACCESS.2021.3084050.

Kurt Hornik Andreas Weingessel Friedrich Leisch Chih-Chung Chang Chih-Chen Lin David Meyer, Evgenia Dimitriadou. e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. https://cran.r-project.org/package=e1071, 2023. [Online; accessed 12-12-2023].

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

Steve Weston Andre Williams Chris Keefer Allan Engelhardt Tony Cooper-Zachary Mayer Brenton Kenkel R Core Team Michael Benesty Reynald Lescarbeau Andrew Ziem Luca Scrucca Yuan Tang Can Candan Tyler Hunt Max Kuhn, Jed Wing. caret: Classification and Regression Training. https://cran.r-project.org/package=caret, 2023. [Online; accessed 12-12-2023].

Mc Hugh ML. Interrater reliability: the kappa statistic. *Biochem Med (Zagreb)*, 22:276–282, 2012.

Robert Gilmore Pontius and Marco Millones. Death to kappa: birth of quantity disagreement and allocation disagreement for accuracy assessment. *International Journal of Remote Sensing*, 32(15):4407–4429, 2011. doi:10.1080/01431161.2011.552923.

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.

Brian Ripley Terry Therneau, Beth Atkinson. rpart: Recursive Partitioning and Regression Trees. https://cran.r-project.org/package=rpart, 2023. [Online; accessed 12-12-2023].