To Higgs or not to Higgs

Project 2 - Statistical Methods in Data Mining

Bruno Parracho and Telmo Monteiro

13th December 2023

Mathematics Department, Faculty of Sciences of University of Porto

Introduction

Objective: Which collisions produce the Higgs

- Comparing the signal process and the background (null hypothesis)

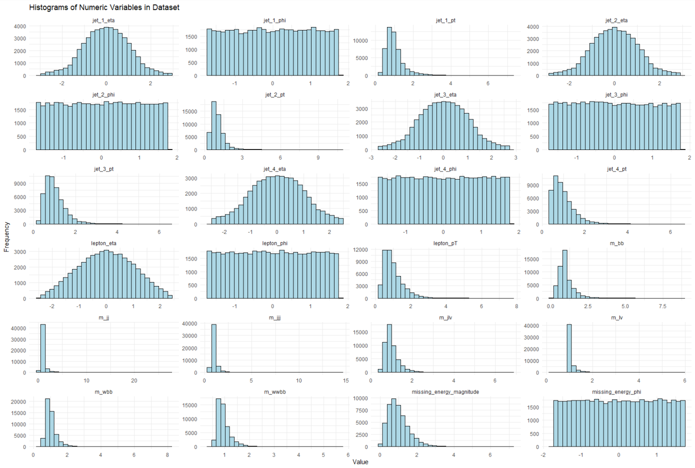


Figure: Histogram of numerical variables of 50 000 randomly chosen events from the original data set.

Data set

- 21 features that are kinematic properties and 7 features that are functions of the first 21. Grand total of 28 features
- Target: $\sim$ 46% for label 0 and $\sim$ 54% for label 1

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
⋮	⋮	⋮	⋮	⋮	⋮	⋮
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

Table: Some basic statistics about the data used.

- Features have different scales

Linear and Quadratic Discriminant Analysis

Objectives

- Reduce the dimensionality of the data
- Similar to PCA, but different

Maximize the posteriori probability, using MLE, by the function

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

- Assumes the priori probability are gaussian
- LDA, the covariance matrix for both classes are equal

Confusion matrices of LDA and QDA

- From R package MASS, the function lda and qda
- 10000 points: training data 70% and test data 30%
- LDA and QDA already scale the data

Table: Confusion matrix for LDA.

	Reference	
Prediction	0	1
0	2410	1257
1	2292	4041

Table: Confusion matrix for QDA.

	Reference	
Prediction	0	1
0	2118	983
1	2584	4315

Statistical quantities

- **Accuracy:** $\frac{TP+TN}{P+N}$
- **Sensitivity:** $\frac{TP}{P}$
- **Specificity:** $\frac{TN}{N}$
- **Cohen's kappa:**

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

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

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

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

Accuracy	**Sensitivity**	**Specificity**	κ
0.6433	0.4504	0.8145	0.2701

Decision Tree

Objectives

- Tree-based methods **partition the feature space** into a set of rectangles
- Fit a simple model (like a constant) in each one
- Conceptually **simple** yet **powerful**
- **CART**: popular method for tree-based classification, applied

Default splitting index used was the default, the **Gini index**, a measure of the node impurity:

$$i(t) = \sum_{i \neq j} p_i p_j = 1 - \sum_{j=1}^{K} p_j^2, \tag{3}$$

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

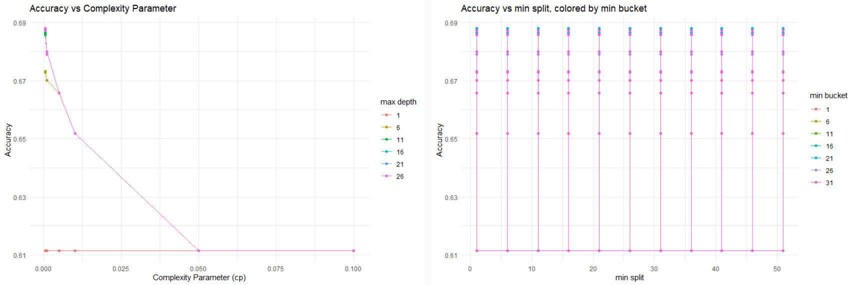
Decision Tree

- R package *rpart*
- **50 000 points**: training data 70% and test data 30%
- **2 scenarios:** original data vs scaled (centered with range bounds of 0 and 1)
- **cyclically** computing the model with train data and corresponding accuracy through the **grid** of hyper-parameters

Table: 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

Decision Tree

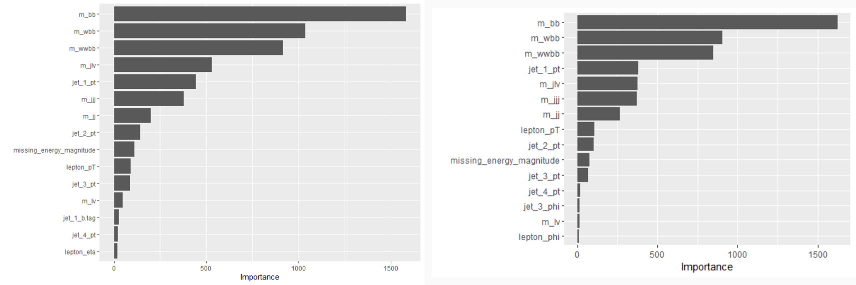


(a) Accuracy as a function of complexity parameter, for different values of depth.

(b) Accuracy as a function of minimum split, for different values of minimum bucket.

Figure: Un-scaled data.

Decision Tree



(a) Un-scaled data.

(b) Scaled data.

Figure: Importance of the first 15 variables.

Decision Tree

Table: Confusion matrices.

Table: Un-scaled data.

	Reference	
Prediction	0	1
0	4604	2251
1	2428	5716

Table: Scaled data.

	Reference	
Prediction	0	1
0	4596	2234
1	2490	5679

Table: 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: 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

Decision Tree

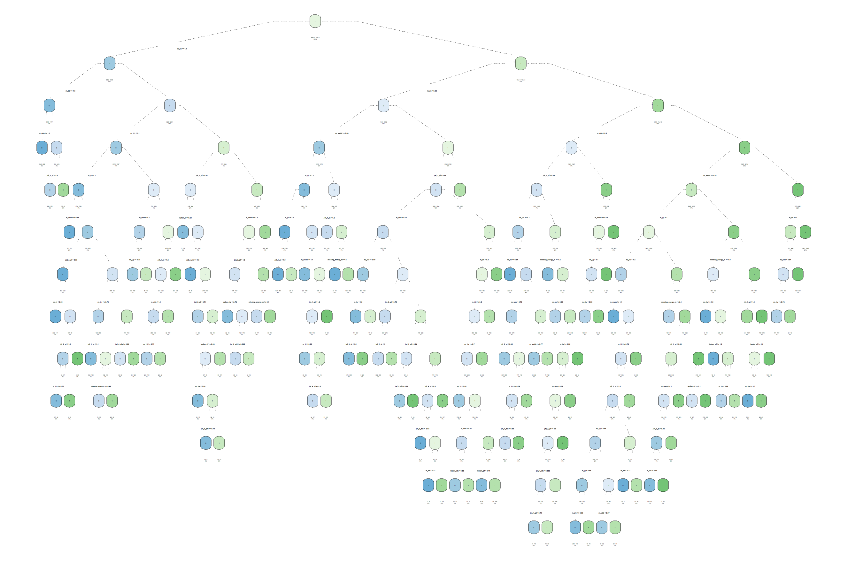


Figure: Scheme of decision tree for un-scaled data, where we can see the splitting rules, the nodes and leafs. Unfortunately, the tree is too deep to effectively visualize it.

k-Nearest Neighboors (kNN)

Objective

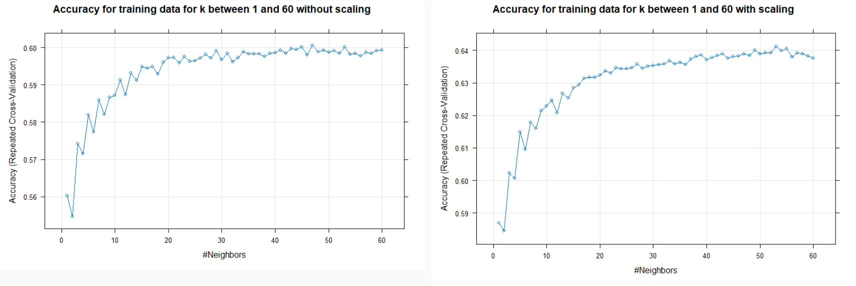
- Use observations in training set closest in input space to x to form $\hat{Y}$
- Specifically, the k-nearest neighbor fit for $\hat{Y}$ is defined as follows:

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i, \tag{4}$$

  where $N_k(x)$ is the neighborhood of x defined by the k closest points x_i in the training sample
- Distance metric used was Euclidean distance
- Find the k observations with x_i closest to x in input space, and average their responses

k-Nearest Neighboors (kNN)

- R package *caret*
- **50 000 points**: training data 70% and test data 30%
- **2 scenarios:** original data vs scaled (centered with range bounds of 0 and 1)
- 10-fold cross-validation, *k* varying from 1 to 60



(a) Accuracy for un-scaled training data using 10-fold cross-validation with *k* ranging from 1 to 60.

(b) Same, but for scaled training data.

Figure: Training accuracies.

k-Nearest Neighboors (kNN)

Table: Confusion matrices.

Table: Un-scaled data.

	Reference	
Prediction	0	1
0	2285	1298
1	4731	6685

Table: Scaled data.

	Reference	
Prediction	0	1
0	3109	1474
1	3970	6446

Table: 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

Support Vector Machines (SVM)

Objectives

- Make the kernel method computational feasible
- Find the best separable hyperplanes
- Boundary to marginalise the margin

Classifier,

$$\hat{y}(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i \mathcal{K}(x, x_i)\right) \tag{5}$$

Support Vector Machines (SVM)

- From R package e1071, the function svm: Radial, Linear and Sigmoid
- From R package kernlab, the function ksvm: Laplace
- 50000 points: training data 70% and test data 30%

Table: Confusion matrix for SVM with a radial kernel

	Reference	
Prediction	0	1
0	4145	1827
1	2905	6122

Table: Confusion matrix for SVM with a linear kernel

	Reference	
Prediction	0	1
0	3237	1516
1	3831	6416

Table: Confusion matrix for SVM with a sigmoid kernel

	Reference	
Prediction	0	1
0	3740	3311
1	3334	4614

Table: Confusion matrix for SVM with a laplace kernel

	Reference	
Prediction	0	1
0	3547	1513
1	3509	6431

Support Vector Machines (SVM)

Table: 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

Conclusions

Conclusions

- **Accuracy** as the defining metric of the performance of the models
- **Every model**, except SVM with sigmoid kernel obtained accuracy > 60%
- **Best model:** decision tree with un-scaled data
- Not good enough results, but they provide a glimpse of the limits of these simple models in this problem: simple answer (binary) - complex process situation (particle physics)

Table: 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

Thank

Thank you!