EURECOM

Sophia Antipolis

Analysis of the MAGIC Gamma Telescope Data Set

Final Project for MALIS course

Professor Maria A. Zuluaga

Simone PAPICCHIO *simone.papicchio@eurecom.fr*
Daniele FALCETTA *daniele.falcetta@eurecom.fr*
Francesco CAPANO *francesco.capano@eurecom.fr*

code: *github.com/spapicchio/Gamma-Telescope-Analysis*

January 20, 2022

Contents

1 Dataset Overview

The data present in this dataset are generated to simulate registration of high energy gamma particles in a ground-based atmosphere. Cherenkov gamma telescope (CTA) observes high energy gamma rays, taking advantage of the radiation emitted by charged particles produced inside the electro-magnetic showers initiated by the gammas, and developing them in the atmosphere. More precisely, when the gamma rays reach the earth's atmosphere they interact with it, producing cascades of subatomic particles. These cascades are also known as air or particle showers. Light travels 0.03 percent slower in air, thus these ultra-high energy particles can travel faster than light in air, creating a blue flash of "Cherenkov light". Although the light is spread over a large area (250 m in diameter), the cascade only lasts a few billionths of a second. CTA's large mirrors and high-speed cameras will detect the flash of light and image the cascade generated by the gamma rays for further study of their cosmic sources allowing reconstruction of the shower parameters. The available information consists of pulses left by the incoming Cherenkov photons on the photomultiplier tubes. Depending on the energy of the primary gamma, a total of few hundreds to some 10000 Cherenkov photons get collected, in patterns (called the shower image), allowing to discriminate statistically those caused by primary gammas (signal) from the images of hadronic showers initiated by cosmic rays in the upper atmosphere (background). The goal of the classification is to correctly classify the gamma signal (g) from hadron (h, background).

2 Dataset Analysis

The first step of the project is the data analysis. We used different representation of data based to the information needed to be extracted. The results are showed in the following paragraph. The dataset is composed of **19020** distinct records with **11** features and does not contain missing values. The following table shows the features.

Attribute	**Type**	**Description**
fLength	*Real*	major axis of ellipse
fWidth	*Real*	minor axis of ellipse
fSize	*Real*	10-log of sum of content of all pixels
fConc	*Real*	ratio of sum of two highest pixels over fSize
fConc1	*Real*	ratio of highest pixel over fSize
fAsym	*Real*	distance from highest pixel to center, projected onto major axis
fM3Long	*Real*	3rd root of third moment along major axis
fM3Trans	*Real*	3rd root of third moment along minor axis
fAlpha	*Real*	angle of major axis with vector to origin
fAlpfDistha	*Real*	distance from origin to center of ellipse
class	*Categorical*	g gamma (signal), h hadron (background)

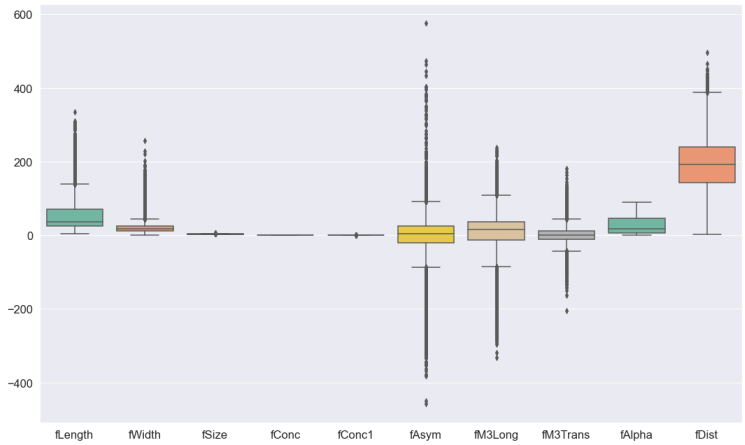

Figure 1: Boxplot of the features compared. Interesting to notice the outliers and the different scale of the data

From the boxplot in Fig. 1 is possible to notice that the range of values for each feature is different. Therefore, in order to not have bias in the model towards higher/lower values, we need to scale the dataset.

Moreover, some "distant" measurements are visible in the Figure 1, but we do not know if they are measurement errors or some possible outliers useful for the classification. To cover all the possibility we performed different experiments with different parameters. See the Method sections for the details.

Crucial, it is also to understand if there is some correlation between features. We used the Pearson correlation measure (value between -1 and 1) which is a measure of linear correlation between two sets of data. It is computed as the ratio between the covariance of two variables and the product of their standard deviations. High ρ correlation means that the two features are linearly correlated, that is, to be related in such a manner that their values form a straight line when plotted on a graph. In Figure 2 is shown, as heatmap, the correlation between features. The features more correlated are **fConc** and **fConc1**. We could reasonably expect this high ρ by their definitions. We can safely remove *fConc1*.

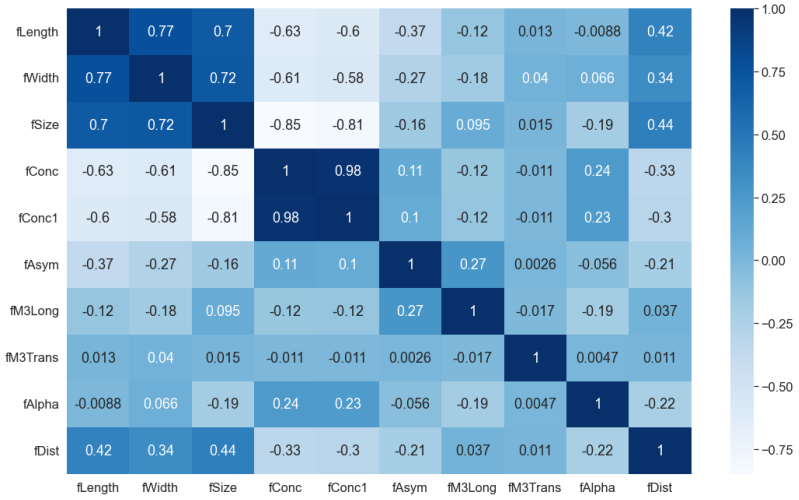

Figure 2: Heatmap of the correlation features using the Pearson correlation.

We have performed an analysis to check the balanceness of the dataset. It is balanced when it contains equal or almost equal number of samples from each class, “unbalanced”, as in this case, if the samples from one of the classes outnumber the others. The Dataset is clearly imbalanced, since there are 12332 **(65%)** samples labelled with **Gamma** and 6688 **(35%)** labelled as **Hadron**. This kind of skewness towards the label gamma may cause problems during the training. Some classifiers, such as Random Forest, fails to address this kind of problem as they are sensitive to the proportions of classes, i.e they tend to favor the class with the largest proportion of observations (majority class).

3 Training Pipeline

Outlier Removal: We use Local Outlier Factor to remove "distant" measure that may be measurement errors or outliers. In addition, we performed several experiments to cover all the possible cases.

Stratified Training/Test split: We use a stratified split to be able to replicate the real world in the test set (Training data 65%, Test Data 35%).

Robust Scaler: Typically, this is performed by removing the mean and scaling to unit variance. However, with the presence of the outliers the sample mean / variance can be influenced in a negative way. In such cases, the median and the interquartile range often give better results.

SMOTE: We have seen that the dataset is imbalanced towards the class Hadron (65% g and 35%h). There are three solutions: Undersampling the majority class (Gamma), Oversampling the minority class (Hadron) or leave the dataset imbalanced. Undersampling the majority class in this case it is not recommended because the goal of the classification is to minimize the error on Hadron since classifying a background event (h) as signal (g) is worse than the opposite. For this reason, it is better to keep as many records as possible with label (Gamma). The third solution has to be excluded since leaving the dataset imbalanced towards g will tend to favor this class respect to h, i.e. leading to more wrong prediction for h. Oversampling the

majority class, instead, seems to be the correct choice for this dataset. The selected algorithm for oversampling is the Synthetic Minority Oversampling Technique (SMOTE).

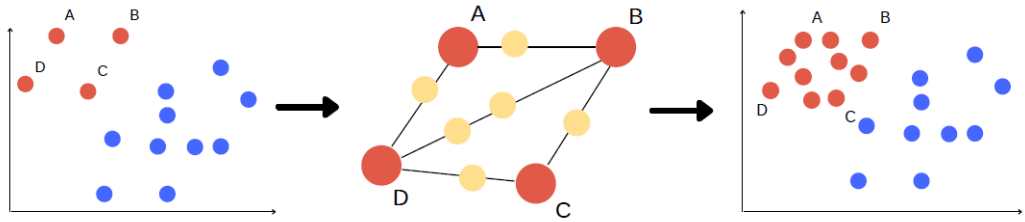

Figure 3: The figure on the left shows an example of imbalance dataset. In the central Figure is shown as SMOTE works (yellow circles are the new samples). The third figure instead represents the balanced dataset after using SMOTE. *Source: Created by us*

Stratified Cross Validation: Crucial is to select a stratified version of the CV. In addition, only the training set is oversampled (see Fig. 4). The selection of **k** is not trivial. Higher value of k will produce a small validation dataset (poor estimation of the true error). Lower value of k will create a small training dataset (poor generalization). For this analysis, k is set to 11.

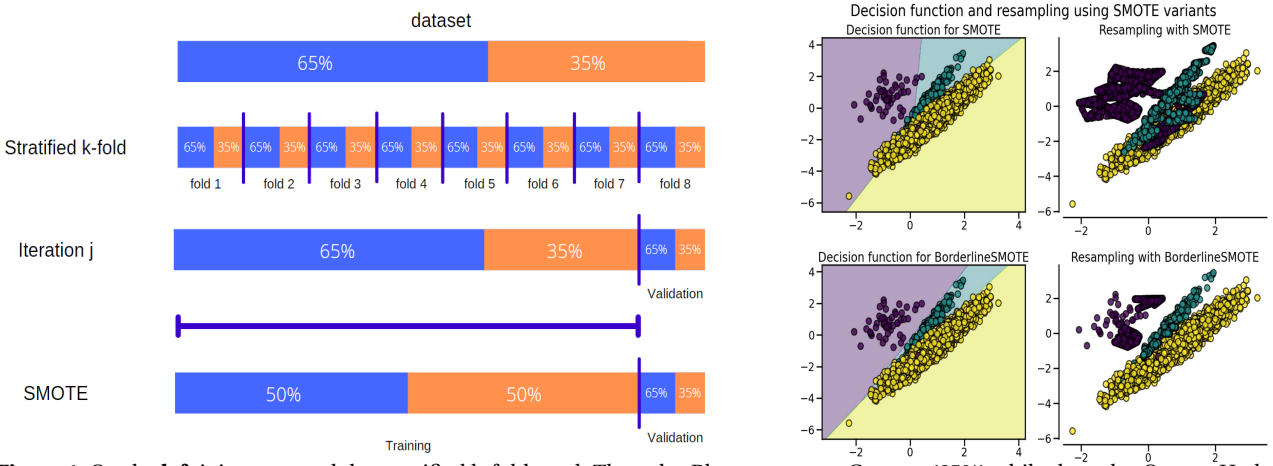

Figure 4: On the **left** it is presented the stratified k-fold used. The color Blue represents Gamma (65%) while the color Orange Hadron (35%). Only the training set is oversampled (using SMOTE). Instead the validation one is untouched. The k in the figure is set to 8 just to have a clearer image. On the **right** is presented the two different versions of SMOTE used. To note that SMOTE connects the inliers to the outliers. Instead, BorderLineSMOTE up-samples only the points close to the border. *Source: sklearn Source: Created by us*

4 Methods

Multiple experiments have been performed to truly understand the importance of each step in the training pipeline. More precisely, the analyzed steps are the **Outliers Removal**, the application of the **Scaler** and the type of **Up-sampling**.

Ablation	Parameter
LocalOutlierFactor	$[2,3,5,10,15,20,False]$
Scaler	$[True,False]$
Up-Sampling	$["BorderLineSMOTE","SMOTE"]$

Model Analyzed
Random Forest
Logistic Regression
KNN
SVM RBF

The **Local Outlier Factor** calculates the locality factor which is given by k-nearest neighbors. Their distance is used to estimate the local density. The samples with lower density than their neighbors are considered outliers. Larger is the parameter, lower outliers are detected. We also performed analysis with no outliers removal (Parameter = False).
For the **Up-sampling** technique we also decide to analyze a different version of SMOTE. It may connects inliers and outliers which may lead to a sub-optimal solution. BorderLineSMOTE, instead, will detect borderline samples which will be used to generate new synthetic data (see Fig. 4 on the right).
To running the experiments a simple nested loops is used in order to cover all the possibilities and then, for each model, is used a random grid search.

Random Forest
n_estimators → [101, 151, 200]
criterion → [$gini, entropy$]
max_depth → [10, 15, 20, 25, 30, 35]

Logistic Regression
penalty → [$l1, l2$]
C → [0.02, 0.05, 0.08, .1, .15, .2]

KNN
n_neighbors → [30, 35, 38, 50]
weights → [$uniform, distance$]

SVM
C → [.001, .01, .1, 1, 2, 5, 7, 10]
gamma → [$auto, scale$]

There is a common path for the model results. BorderLineSMOTE enable the models to reduce the errors on Hadron (Hadron classified as Gamma) but increasing the error on Gamma. SMOTE, instead, reduce the error on Gamma (Gamma classified as Hadron) with an increasing error on Hadron.
The general results on the outliers and the scaler are reported in the table below:

Model	Scaler = $True$ ⟶False	Neighbors = 2 ⟶ $False$
Random Forest	Invariant	Invariant
Logistic Regression	Invariant	Invariant
KNN	Decreased Prediction	Increased Prediction
SVM RBF	Decreased Prediction	Decreased Prediction

The **Random Forest** and the **Logistic Regression** are robust to the outliers and to the scale of the data. Indeed, no relevant changes are detected. The **KNN** is sensible to the scaler, but apparently, for this dataset it needs the outliers. The other way around for the **RBF SVM** which is sensible to both the not scaled data and the outliers.

5 Results and Conclusion

For each experiment we have used a Random Grid Search to catch the best model hyper-parameters. We evaluate our models using the F1 score and not the accuracy, which can be misleading with imbalanced data. In addition, it gives a better measure of the incorrectly classified particles. The best models are selected according to the rate of misclassified hadron samples (Fig. 5). The simple classification accuracy is not meaningful for this data, since classifying a background event as signal is worse than classifying a signal event as background. Therefore, lower error on Hadron means a better model.

Model	Hadron Misclassified rate (Borderline SMOTE)	Hadron Misclassified rate (SMOTE)
SVM RBF	0.14	0.19
RandomForest	0.16	0.17
KNN	0.18	0.25
Logistic Regression	0.23	0.27

The previous table reports the best misclasified rate of Hadron for the analyzed models. Even though, the kernelized SVM has the lowest rate, the random forest performs better on the error on Gamma (see fig. 5). So at the end we can state that they are the two best models. In the table below are reported the best hyper-parameters.

Random Forest	n_estimators → 200	criterion → $entropy$	max_depth → 20
SVM	C → 10	gamma → $scale$	

To be noted that the best results are obtained removing the outlier and using the scaler (as shown in the labels in Fig. 5).

In order to understand if we could go further with the accuracy, we also plot the learning curves (fig. 6). Even though we tried to reduce as much as possible the overfitting by means of the max_depth hyper-parameter tuning , the training score for the Random Forest is always one. This is caused by the high complexity of the model. A possible way to reduce overfitting may be to increase the training dataset.

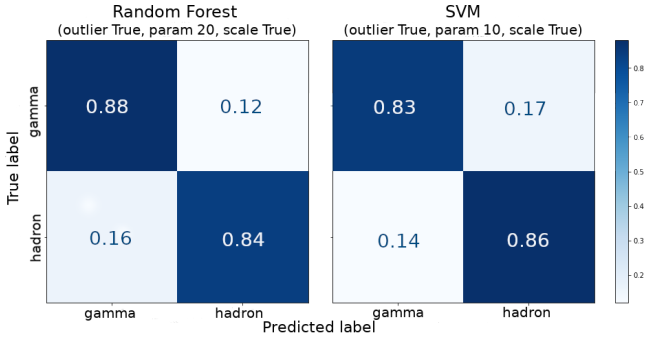

Figure 5: Confusion matrices for the RBF SVM and the Random Forest (both using Bordeline SMOTE)along with the best parameters for the experiments (scaler, outlier, outlier parameters)

The learning curve for the SVM instead shows that the model as obtained almost its maximum potential. Even with other data, the training score and the test score would converge around 0.85. Therefore, in order to improve its score we should try a more complex model.

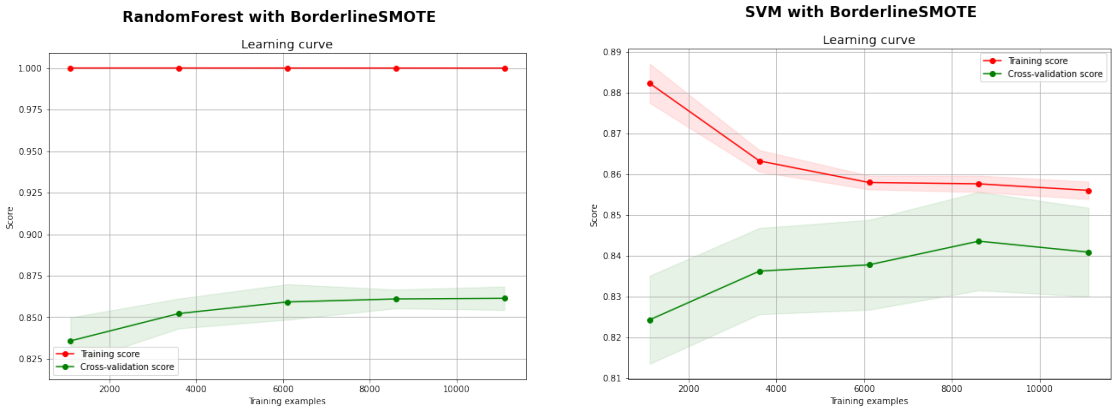

Figure 6: Learning curves using the best performing set of parameters for each model

For the sake of completeness, are reported the ROC curves for all the models. It is possible to notice that the area under the curves is, almost for each model, higher than 0.9. The logistic regression has the worst performances.

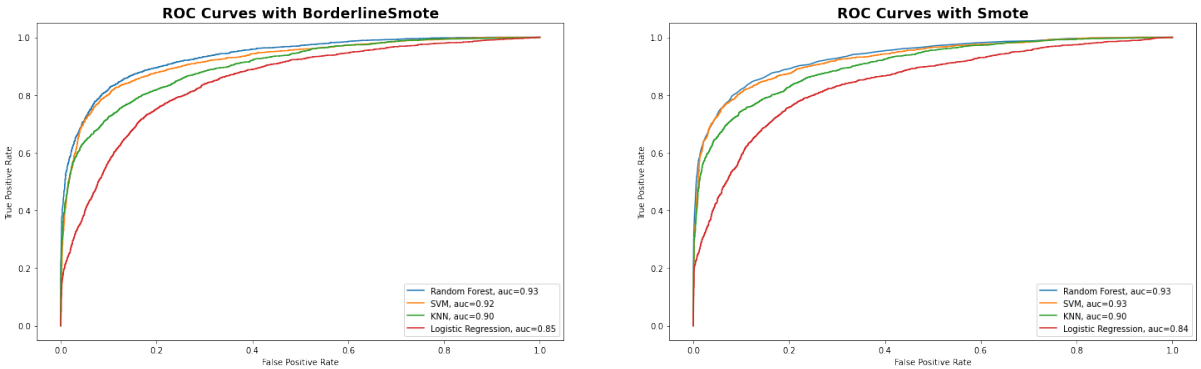

Figure 7: This figures shows the ROC curves for each model with the best set of hyper-parameters

To note that all the models got higher performances with the usage of Borderline SMOTE since oversampling in the border allows to distinguish in a better way the Hadron particles.

SP: Define/code the training pipeline (Sec. 3). Define/code the experiments (Sec. 4). Written the processing_model notebook (the code for the experiments and the best model pipelines.)
FC: Define/code the dataset preliminary analysis and visualizations (Sec. 2).
DF: Define/code the tested models and hyperparameters (Sec. 4). Written the corresponding part in the processing_model notebook.
The part regarding defining/coding the results analysis (Sec. 5) and the Curves notebook (contains graphs such as the learning and the ROC curves) jointly developed by **FC** and **DF**.