Higgs Boson

Group: Machine_learnia2: Paul Armand Douglas Bouchet, Adam Chellaoui, Marina Rapellini

1 Introduction.

This project analyzes a dataset and its features using machine learning methods in order to be able to extract meaningful information and to generate predictions using those methods.

The dataset is studied in order to be able to analyze the products that result from the collision of protons at high speeds; the only way to be able to tell if these collions have produced a Higgs boson (whose decay is too fast to let the particle be observed) is by studying the likelihood that a given event's signature (the products resulted from the decay process) was the result of a Higgs boson (signal) or some other process/particle (background).

A vector will represent the decay signature of a collision event and the models and methods used are going to determine if this event was signal (a Higgs boson) or background (something else) using binary classification.

2 Models and Methods.

Before diving into analyzing the dataset, the latter has to be standardized in order to be able to forget about the unit and to be able to compare the data. While importing the data (each data is imported with its identifier and its label), the meaningless values of -999 are replaced by the median value of the column (feature) to which it belongs; also the outliers are deleted from the data set by choosing a maximum and a minimum threshold in order to remove the 1/1000 highest and lowest values. As we need to consider the submission-test data as unseen, we standardize the data with the mean and the standard deviation of the training model features.

The value of $\boldsymbol{w}$ has to be calculated in order to predict the value of $\boldsymbol{y}$ given by $\boldsymbol{y} = \boldsymbol{x}^T\boldsymbol{w}$ if linear regression for classification is used or $\boldsymbol{y} = \sigma(\boldsymbol{x}^T\boldsymbol{w})$ for logistic regression; its value is first computed using a training set and it is later used to predict the values of the test set ; different methods can be used in order to calculate $\boldsymbol{w}$.

The value of the error has to be minimized; this can be done using the gradient descent method which uses the full gradient vector (with a complexity $\mathcal{O}(N \times D)$) or the stochastic gradient whose calculation is restricted to a random subset of the training set (with a reduced complexity $\mathcal{O}(B \times D)$ with B the size of the batch generated) and whose convergence is going to be slower. Another way to compute $\boldsymbol{w}$ for linear regression, is by using a mean-squared error. It is solving analytically a linear system of equations called the normal equation whose solution is given by $\boldsymbol{w}^*=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.

The data set is split using cross-validation in the optimization phase for determining the best parameters; data is randomly divided into K groups and trained K times, each time K-1 set are used for training and 1 set for testing; the average of the K results is then used.

In order to increase our convergence and to reduce the risk of underfitting the input is augmented by polynomial feature expansion, and regularized with a λ parameter to prevent it from overfitting.

Ridge regression. In order to avoid over-fitting, ridge regression is implemented; this regularization consists in adding a term called the regularizer (in this case the standard Euclidian norm $\lambda||\boldsymbol{w}||_2^2$) in order to penalize complex models and favor simpler ones (the error that is calculated is always MSE); by using cross-validation, the best values for the hyper-parameters (which for us are going to be the degree of the polynomial expansion and the parameter λ) are determined.

Logistic regression. The method of logistic regression is used when considereing a binary classification; it uses the sigmoid function in order predict the (posterior) probability of the two class labels. At the beginning, the polynomial expansion for the logistic regression could not be implemented because the value of the sigmoid function was going towards infinity; in order to avoid this problem and to be able to obtained an extended feature vector, the sigmoid function was re-defined as: $\sigma(t) = 0.5 * (1+\tanh(0.5t))$ and the function that computes the cost by negative log-likelihood was modified by adding a value of $\epsilon = e^{-5}$ which is added to the value predicted using the sigmoid function when calculating the loss.

3 Results.

The best results are obtained using the ridge-regression model; our work was then focused on optimizing the training of our model using the latter method. To do so, a function for finding the optimal hyperparameters is implemented (best_parameters_ridge_regression) that performs cross-validation steps in order to find the optimal lambda and degree for polynomial feature expansion.

The highest value of accuracy (0.803) is obtained by implementing ridge-regression on the extended feature vector and by using cross-validation with 10-fold, degrees $\in [1, 5]$ and lambdas $\in [10^{-6}, 10^{-4}]$. We chose this value in order to have a better granularity and reduce the variance between the results, although it is more expensive when considering computational time. We obtained degree = 5 and $\lambda = 3.0538^{-5}$.

The values that were obtained by implementing logistic regression with a polynomial expansion of degree 2 and $\lambda = 10^{-4}$, always using cross-validation with K-fold = 4, were around 0.78; a value of 0.71 was obtained when not implementing the polynomial expansion.

Our best model got the following results on the testing set submitted to Aicrowd: **0.803 of categorical accuracy**, and **0.694 of f1 score**, meaning that our model actually find some correlations between features and Higg's boson prediction.

A similar function was implemented also for the logistic regression but the obtained model was not as good as the one obtained with ridge regression; the first one uses a second derivative in order to update the value of $\boldsymbol{w}$ ($\boldsymbol{w}^{(t+1)} := \boldsymbol{w}^{(t)} - \gamma^{(t)}\ (\mathbf{H}^{(t)})^{-1}\nabla\mathcal{L}(\boldsymbol{w}^{(t)})$) so each step takes a longer time (also when stochastic gradient descent is implemented) making trying to optimize this method not rewarding.

4 Discussion.

An idea to possibly improve our result was using a function called equalize_classes in order to be able to balance the percentage of bosons and background that is present in our data; this is done in order to avoid a high percentage of only one type of data and thus having an unbalanced model which can have an effect of favoring one class, especially when using ridge-regression for classification. But there was not a significant change in our results.

It is also interesting explaining why removing outliers is useful; a possibility in order to justify this could be perfoming the following: making a statistical study of the data to see if it follows a gaussian distribution; if that is the case, then removing outliers (using for example the 3-σ limits) would make sense because some parts of the data will be far away from the mean. But if the data don't follow a gaussian distribution, then it is likely that, even if 0.1% of the extreme values is removed, many extreme values will still be present.

References

[1] Slides, weeks 1-2-3-4-5