Detection of Higgs Boson Events in High-Energy Physics Using Ensemble and Neural Network Models

1 Objective

The discovery of the Higgs Boson at the Large Hadron Collider (LHC), CERN, Geneva, was acknowledged by the 2013 Nobel Prize in Physics. However, many properties of this newly discovered particle remain unknown and require further dedicated study at the LHC. Therefore, the objective of this challenge is to improve upon the standard LHC analysis using Machine Learning (ML) techniques. I use XGBoost / Gradient Boosting / Random Forest ensembles / Neural Network to train the dataset and finally perform AMS threshold optimization for signal detection.

2 Significance of Discovery

- High-energy proton-proton collisions at the LHC produce thousands of particles in each event. Events that satisfy certain criteria are recorded, generating petabytes of data annually.
- In particle physics, when searching for a rare event like the Higgs boson (signal), we are trying to detect a small "bump" in a sea of already known physical processes (background).
- Among the recorded events, the vast majority correspond to already known physical processes, referred to as **backgrounds**.
- We are primarily interested in discovering new particles or phenomena, which manifest as a significant excess of events compared to the expected background. These are referred to as **signals**.
- Once a region of interest is defined, a statistical (counting) test is performed to evaluate the significance of the observed excess. If the probability that the excess is due to background fluctuations falls below a predefined threshold, the new particle is considered to be discovered.
- Goal is to optimize the selection of **signals** from the **backgrounds**.
- Number of events that come from already known processes, background= B.

- Number of events predicted or observed that are believed to come from the new phenomenon, signal= S.
- significance of discovery= $S/\sqrt{B}$.
- $S/\sqrt{B}$ is true for real life physics experiments with $B \gg S$ (and $B \gg 1$).
- In the simulated data we are working on, since the data contains a lot more fractions of signals than typically realized in nature, a better "significance of discovery" for this problem, a modified function for the significance of discovery, will be taken to be **AMS** (approximate median significance):

$$\text{AMS} = \left\{ 2 \left[(s + b + b_R) \left(1 + \frac{s}{b + b_R} \right) - s \right] \right\}^{1/2}. \tag{1}$$

s = expected number of signal events selected by the classifier, b = expected number of background events selected by the classifier, $b_R = 10$ is set (a regularization term).

3 Setup

- Simulated data: `https://www.kaggle.com/competitions/higgs-boson/data`.
- The data has 30 features: various physical parameters. See Fig. 1 for Correlation Matrix.

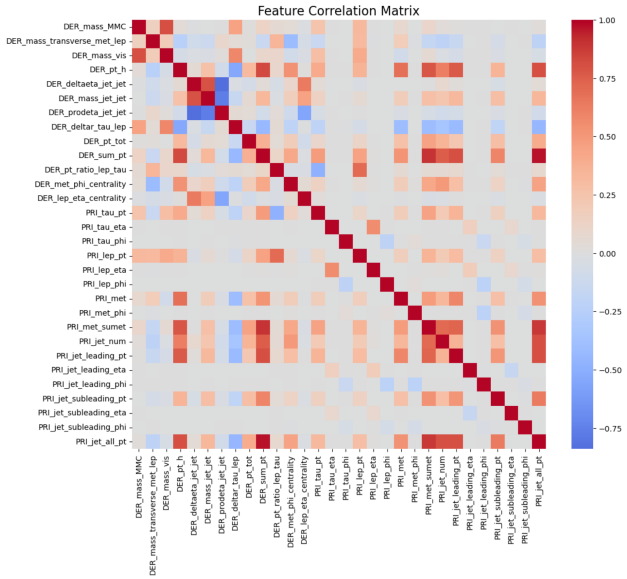

Figure 1: Correlation Matrix.

- The second last item is "weight"; which is not a feature. Hence, weight is not used during training.

- The very last column is the "label/target". Signal/background label.
- Since the dataset has event 'weights', once a training is completed, s, b are computed using their corresponding given weights via:

→ s: the total weighted signal events that were correctly predicted as signal.

→ b: the total weighted background events that were incorrectly predicted as signal.

- To optimize model performance for the AMS metric, I iteratively tested multiple classification thresholds between 0 and 1. For each threshold, I computed the weighted counts of correctly classified signal and background events and calculated the corresponding AMS score. The threshold that yielded the highest AMS was selected as the optimal decision boundary.

4 Strategy-01: Logistic Regression

- First, I have tried the simple Logistic Regression model and obtained an Accuracy=0.75. Not good enough.
- The corresponding Confusion Matrix, see Fig. 2. 9% events are identified as signals, which are actually background. Not good for discovering new physics.

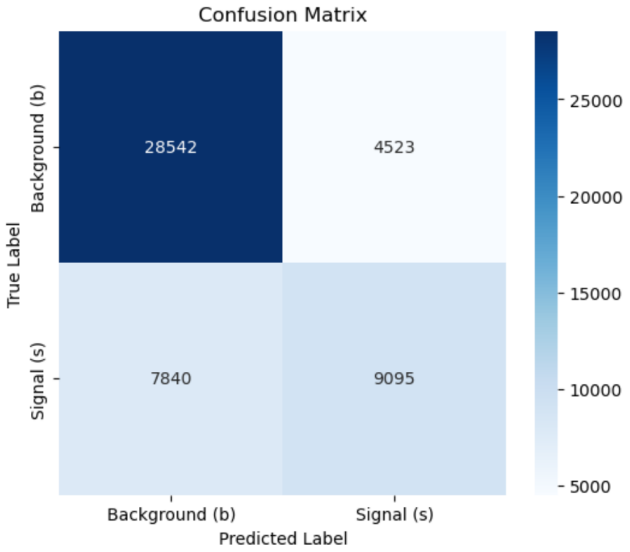

Figure 2: Confusion Matrix: LR.

- Using the trained prediction, we now use the provided "weights" to compute the AMS. Using this AMS formula, we obtain the best AMS value corresponding to a **threshold**=0.42. See Fig. 3, the situation gets even worse as 13.1% events are now identified as signals.

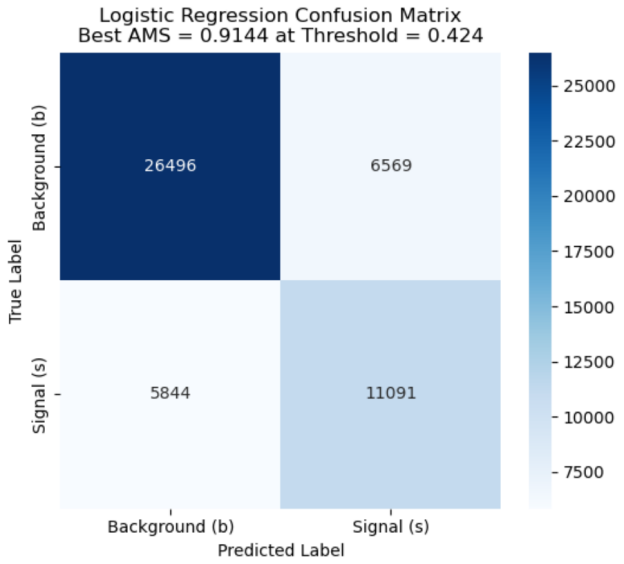

Figure 3: Confusion Matrix: LR after optimizing to best AMS.

5 Strategy-02: Multiple Classifiers test

- Let us now perform a 5-fold stratified cross-validation for **multiple classifiers** (Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and Neural Network). For each fold, train the model (see the Code/Jupyter notebook for the choice of hyperparameters), predict probabilities on the test set, compute the ROC-AUC, and find the mean and standard deviation of the ROC-AUC scores across folds. This allows a fair comparison of models in terms of their ability to separate signal from background, independent of a specific classification threshold.

Table 1: Cross-Validated ROC-AUC Scores for Different Models

Model	**ROC-AUC**	**Std. Dev.**
Logistic Regression	0.8082	0.0018
Random Forest	0.8992	0.0009
Gradient Boosting	0.9090	0.0008
XGBoost	0.9108	0.0010
Neural Net	0.9054	0.0015

- Given that XGBoost obtains the highest mean, we further analyze predictions drawn from this training.
- The corresponding Confusion Matrix, see Fig. 4. 6.6% events are identified as signals, which are actually background. Obviously much better than LR predictions.
- Finally, the best AMS value corresponding to a **threshold**=0.838 is obtained. See Fig. 5, the situation gets much much better as, now, only 0.97% events are identified as signals which are actually backgrounds.

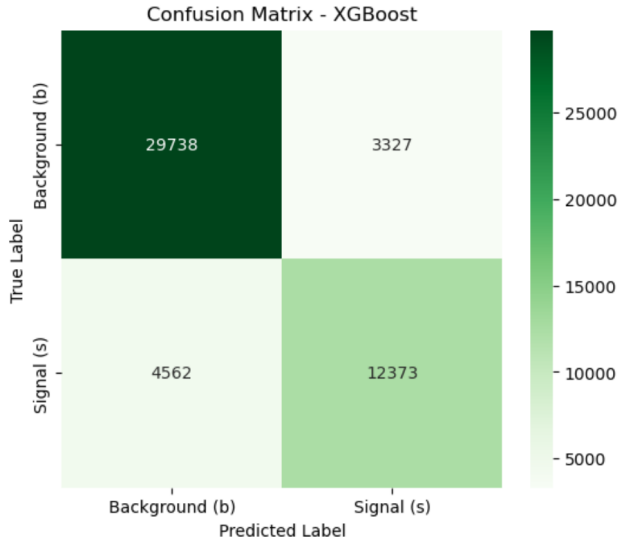

Figure 4: Confusion Matrix: XGBoost.

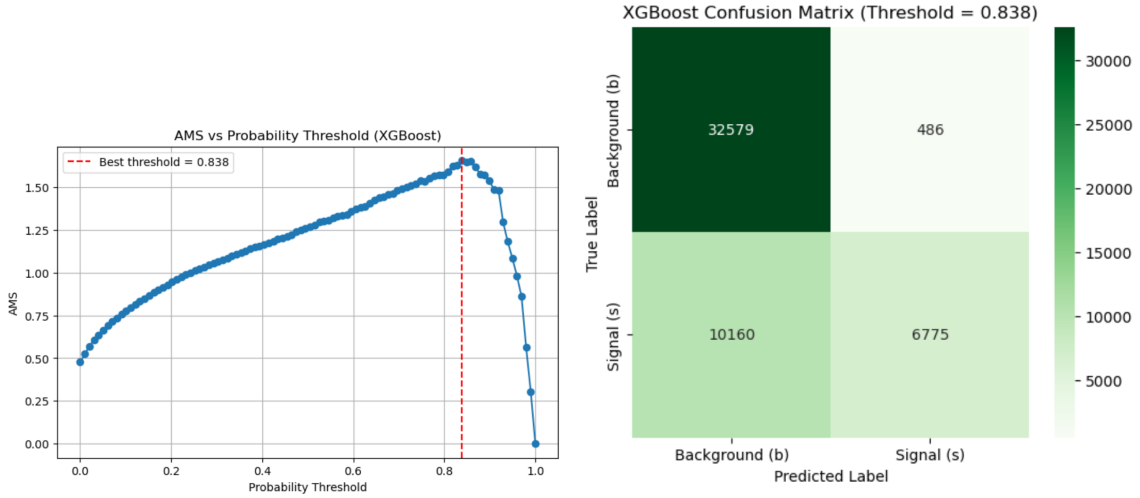

Figure 5: Confusion Matrix: XGBoost after optimizing to best AMS.

6 Skills used

Data Handling & Analysis

- Handling a large, high-energy physics dataset with imbalanced classes (signal vs background).
- Feature scaling and preprocessing for numerical stability and effective model training.
- Event weighting and computation of Approximate Median Significance (AMS) for physics-specific evaluation.
- Model evaluation using ROC-AUC, confusion matrix, and threshold optimization for maximum significance.

ML Modeling

- Building and comparing multiple classifiers: Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and Neural Networks.
- Probability-based threshold optimization to maximize AMS score.
- Hyperparameter tuning for tree-based models and neural networks for improved predictive performance.
- Cross-validation (Stratified K-Folds) to ensure robust evaluation and mitigate overfitting.

Visualization & Communication

- Plotting confusion matrices, ROC curves, AMS vs threshold curves for model performance analysis.
- Visual comparison of multiple ML models with mean $\pm$ standard deviation ROC-AUC.
- Clear presentation of model results and decision-making metrics for real-world high-energy physics applications.

Tech Stack

- Python, Pandas, NumPy, scikit-learn, Logistic Regression, Random Forest, Gradient Boosting, XGBoost, and Neural Network, Matplotlib, Seaborn, SciPy