Signal vs Background Events in High-Energy Physics

Names & student numbers: Tolga KUNTMAN (r0917093) & Michel MOUSSALLY (r0913670)
Assigned coach: Weiyi

Abstract

High-energy physics experiments produce extremely large datasets in which events of physical interest are difficult to distinguish from background noise. Effectively addressing this challenge requires machine learning models that can capture complex patterns in high-dimensional data. In this project, we investigate a binary classification problem using the HIGGS dataset that is a widely used benchmark composed of simulated proton–proton collision events represented by 28 numerical features.

We perform an independent implementation and methodological comparison of three representative machine learning model families such as logistic regression as a linear baseline, gradient-boosted decision trees as a nonlinear ensemble method, and a feed-forward neural network as a deep learning approach. A balanced subset of 500,000 events is sampled from the original dataset of approximately 11 million events to reduce computational complexity. All models are trained and evaluated using a consistent training, validation, and test pipeline, with performance primarily assessed using the area under the ROC curve (ROC-AUC score).

The results show that nonlinear models significantly outperform the linear baseline, highlighting the importance of modelling nonlinear feature interactions in this domain. Gradient-boosted trees and the neural network achieve comparable performance, with only a marginal advantage for the neural network, reflecting diminishing returns from increased model complexity under constrained data and computational resources. Overall, this study provides insight into the trade-offs between model expressiveness, inductive bias, and practical feasibility when applying machine learning methods to large-scale scientific datasets.

1. Introduction

Modern high-energy physics experiments, like the ones that are conducted at CERN's Large Hadron Collider (LHC), generate massive volumes of collision data, in which only a small fraction of data corresponds to physically meaningful signal events [1]. Identifying these rare signal events such as detecting Higgs bosons among background noise is a central challenge in experimental particle physics [2]. The complexity and high dimensionality of detector measurements make this task difficult for traditional statistical techniques, which are often limited in their ability to model nonlinear feature interactions [2]. Therefore, machine learning has become an essential tool in this domain since it offers flexible and data-driven models that are able to learn complex decision boundaries directly from data [2]. The HIGGS dataset, introduced by Baldi *et al.* [3], provides a large-scale and realistic benchmark for studying this

problem. It consists of simulated proton–proton collision events described by 28 continuous features derived from kinematic measurements and physical transformations, with a binary label indicating signal or background.
In this project, we study the binary classification problem on the HIGGS dataset and focus on an independent implementation and methodological comparison of three models which are a linear baseline (logistic regression), a nonlinear ensemble method (gradient-boosted decision trees), and a feed-forward neural network (multilayer perceptron). The input to our models is a vector of 28 numerical features for each collision event, and the output is a predicted probability that the event corresponds to a Higgs signal [4]. By comparing model behaviour and performance using consistent training and evaluation procedures, we aim to gain insight into when complex nonlinear models provide meaningful advantages over simpler alternatives in large-scale scientific datasets.

2. Methods

2.1 Dataset and Features

We use the HIGGS dataset which is a large-scale benchmark dataset [4] for studying machine learning methods in high-energy physics. The dataset is publicly available through the UCI Machine Learning Repository and can be freely accessed for research and educational purposes. It was generated using detailed Monte Carlo simulations of proton-proton collisions at the Large Hadron Collider, with the goal of distinguishing Higgs boson signal events from Standard Model background events.

The purpose of the dataset is to provide a realistic and challenging classification problem that reflects the conditions encountered in modern particle physics experiments. Each data sample corresponds to a single simulated collision event instant and is represented by a vector of numerical features derived from detector measurements and physics-based transformations. The target label indicates whether the event corresponds to a Higgs boson signal (label = 1) or background noise (label = 0).

The dataset contains two classes, which are approximately balanced in our sampled subset, with about 52.9% signal events and 47.1% background events. This moderate imbalance reflects realistic experimental conditions while avoiding extreme class skew that could bias evaluation metrics.

The learning task is a binary classification problem, where the objective is to predict the class label (signal vs. background) from the observed features. In our experiments, we use a randomly sampled subset of 500,000 events with the same label ratio (52.9% to 47.1%) from the full dataset of approximately 11 million events, which allows us to balance computational feasibility with statistical representativeness.

Each event is described by 28 continuous numerical features. These features include 21 raw kinematic measurements (e.g. particle momentum and angular variables) and 7 engineered physics features (e.g. invariant masses, angular separations, and event-level quantities).

The data is provided in CSV format, where each row corresponds to one collision event and each column corresponds to a feature or the binary label. Minimal preprocessing was required since all features are already numerical and standardized in the dataset. We did not apply additional feature engineering or transformations. For models that are sensitive to feature scaling (logistic regression and neural networks), standard normalization was handled internally by the learning algorithms or preprocessing pipelines.

We split the dataset into training, validation, and test sets using an 80/10/10 split [3], ensuring that class proportions remain nearly identical across all splits. The training set is used for model fitting, the validation set for hyperparameter tuning and model selection, and the test set for final performance evaluation.

2.2 Models

To study the impact of model complexity on Higgs boson classification, we evaluate the following models: logistic regression, gradient-boosted decision trees, and a feed-forward neural network (MLP).

2.2.1 Logistic Regression

Logistic regression serves as a linear baseline model and is adapted from the lab exercises. It models the conditional probability of an event being a Higgs signal as

$$P(y = 1 \mid \mathrm{x}) = \sigma(\mathrm{w}^{\mathsf{T}}\mathrm{x} + b)$$

where σ denotes the sigmoid function, x is the 28-dimensional feature vector, and w and b are learnable parameters. The model is trained by minimizing the binary cross-entropy loss using gradient-based optimization. We implemented both a custom gradient descent version (from the lab assignments) and a reference implementation using scikit-learn library [5]. L2 regularization is applied to control overfitting, although the model's capacity remains limited due to its linear decision boundary.

2.2.2 Gradient-Boosted Decision Trees

Gradient-boosted decision trees (GBDT) is a well-suited method for tabular data, which was introduced by Friedman [6]. The model builds an additive sequence of decision trees [7], where each new tree is trained to correct the residual errors of the previous ensemble. The prediction function is

$$f(\mathrm{x}) = \sum_{m=1}^{M} \gamma_m \, h_m(\mathrm{x})$$

where h_m are weak learners (shallow trees) and γ_m are their weights. Key hyperparameters such as the number of trees, tree depth, and learning rate are tuned using the validation set. We

avoid overfitting by using shallow tree depth, learning rate shrinkage, and early stopping based on validation performance. GDBT is trained and tuned by using the XGBoost library [8].

2.2.3 Feed-Forward Neural Network (MLP)

The most expressive and complex model in our study is a fully connected feed-forward neural network, inspired by the architecture introduced by Tomasini [9] and proposed by Baldi et al. [1]. The network consists of five hidden layers, each with 300 neurons, using ReLU [10] activations. The output layer applies a sigmoid activation to produce a probability estimate. The network is trained using the binary cross-entropy loss function, which is standard for probabilistic binary classification tasks [11], and optimized using stochastic gradient-based methods [12]. To reduce overfitting, we used early stopping based on validation AUC [13] and standard regularization techniques. The depth and width of the network allow it to capture complex nonlinear feature interactions that are inaccessible to linear models. The feed-forward neural network models were implemented, trained, tuned, and evaluated using the PyTorch deep learning framework [14].

2.2.4 Hyperparameter Tuning and Validation

Hyperparameter tuning is performed by systematically varying key model-specific hyperparameters and selecting the configuration that maximizes validation performance, measured by ROC-AUC following standard model selection practices in machine learning [15].

- For logistic regression, tuning focuses on the regularization strength (L2 penalty) and optimization settings, balancing bias and variance while preventing overfitting.
- For gradient-boosted decision trees, we tune the number of boosting iterations, maximum tree depth, and learning rate, as these parameters control model capacity, convergence speed, and overfitting [6]. We use early stopping based on validation performance to stop training when additional trees no longer improve generalization [13].
- For the feed-forward neural network, we tune architectural and training parameters such as network depth and width, learning rate, and early stopping. Early stopping is the primary regularization mechanism, preventing overfitting by stopping training when the validation ROC-AUC stops improving [13].

2.2.5 Evaluation Metrics

Model performance is evaluated primarily using the Area Under the Receiver Operating Characteristic Curve (ROC-AUC). ROC-AUC measures a classifier’s ability to distinguish signal from background across all possible decision thresholds, making it well suited for rare-event classification problems in high-energy physics [3].

3. Results

All the models were evaluated on the same test set using the ROC-AUC, and table 1 summarizes the ROC-AUC scores for all three models.

Model	AUC
Logistic Regression	0.661
Gradient Boosted Trees	0.824
Feed-Forward MLP	0.828

Table 1: AUC Comparison for All the Trained Models

The results show a clear performance gap between the linear baseline and the nonlinear models. Both gradient-boosted trees and the MLP outperform logistic regression, with the MLP achieving the highest ROC-AUC, by a small margin.

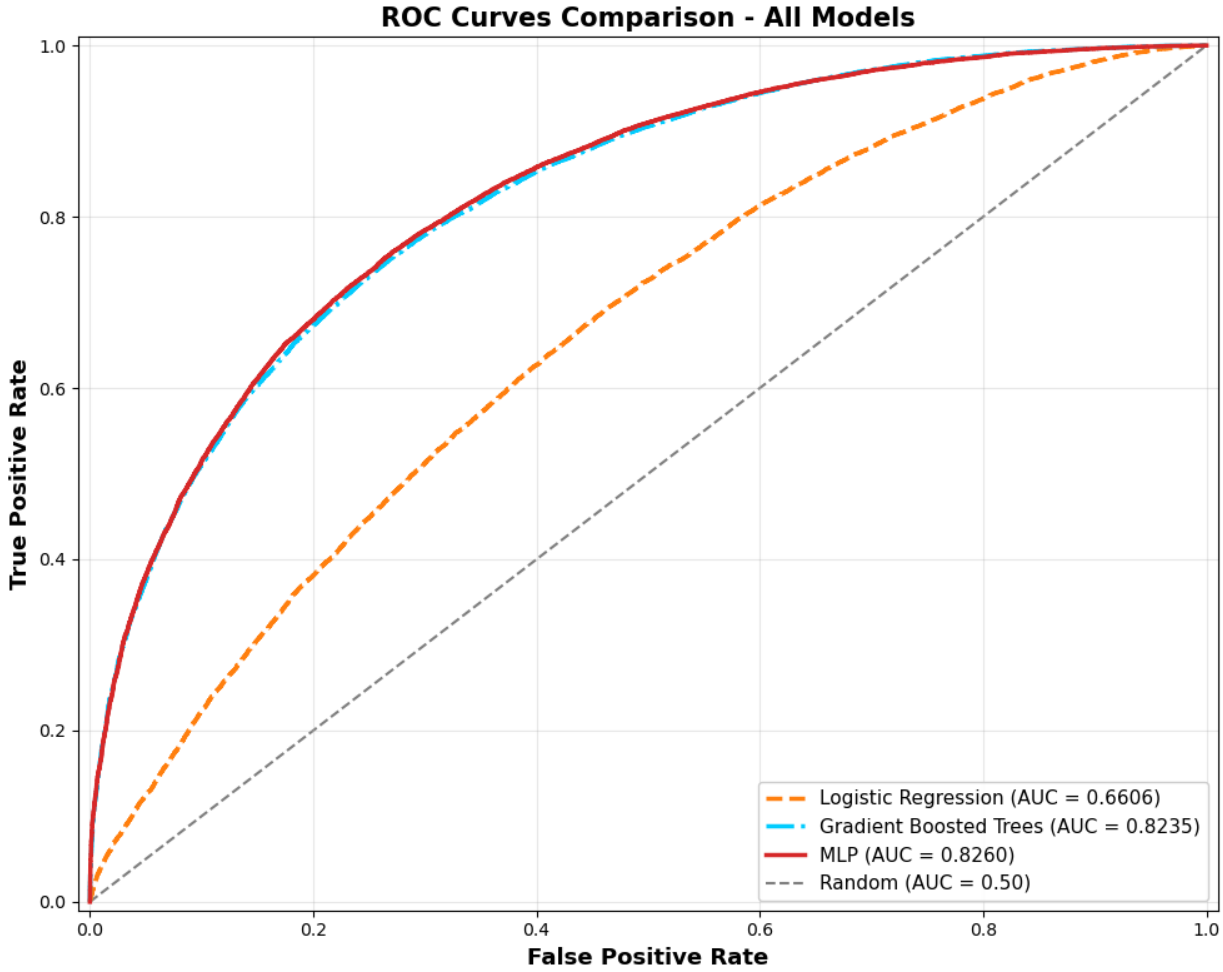

Figure 1: ROC Curves Comparison for All the Trained Models

Logistic regression serves as a linear baseline for the classification task. On the test set, it

achieves a ROC-AUC of 0.661, indicating limited discriminative power when constrained to a linear decision boundary in the 28-dimensional feature space.
The ROC curve shown in figure 1 shows moderate separation between signal and background but remains close to the diagonal compared to the nonlinear models, reflecting limited ranking capability.
The gradient-boosted decision tree model significantly improves classification performance, achieving a test ROC-AUC of 0.824. This improvement reflects the model's ability to capture nonlinear feature interactions and localized decision boundaries in tabular data.
Compared to logistic regression, the ROC curve in figure 1 is consistently higher across the full range of false positive rates, indicating stronger signal-background discrimination.
The feed-forward neural network achieves the highest performance, with a ROC-AUC of 0.828. The improvement over gradient-boosted trees is marginal but consistent.
The ROC curve as shown in figure 1 is very similar to that of the gradient-boosted trees.

4. Discussion

4.1 Interpretation of Findings

The experimental results show a clear and consistent performance gap between the linear baseline and the nonlinear models on the HIGGS binary classification task. Logistic regression achieves a ROC-AUC of approximately 0.66, while gradient-boosted decision trees and the feed-forward neural network reach significantly higher values, around 0.82–0.83. This outcome matches the expectations for high-dimensional physics data, where complex and nonlinear feature interactions play a crucial role in separating signal from background events [3].
The relatively weak performance of logistic regression can be attributed to its linear decision boundary, which limits its ability to exploit nonlinear dependencies among the 28 input features. While L2 regularization helps control overfitting and stabilizes training, it does not increase the representational capacity of the model. As a result, logistic regression appears to be capacity-limited, capturing only coarse trends in the data.
Both gradient-boosted decision trees and the MLP outperform the linear baseline. Gradient boosting is well known to perform strongly on tabular datasets by modelling localized nonlinear patterns through an ensemble of shallow trees [6]. The feed-forward neural network achieves the highest ROC-AUC, but the improvement over gradient-boosted trees is small. This relatively small gap was somewhat unexpected given the higher power and complexity of deep neural networks, but it can be explained by several factors. First, the models were trained on a reduced subset of 500,000 samples rather than the full HIGGS dataset with 11 million samples [4], which limits the extent to which deep architectures can fully leverage their capacity, as also observed in prior work [3]. Second, gradient-boosted trees have a strong inductive bias for structured tabular data and can achieve competitive performance even with limited hyperparameter tuning [16].

The chosen hyperparameters reflect a trade-off between model capacity, generalization, and computational feasibility. Shallow tree depths and learning-rate shrinkage were used for gradient boosting to prevent overfitting. Moreover, early stopping based on validation ROC-AUC was applied across models to ensure fair comparison. For the neural network, early stopping served as the primary regularization mechanism which allowed the model to capture non-linear feature interactions without excessive overfitting [17]. Overall, the observed performance trends meet expectations and indicate that the nonlinear models successfully capture underlying patterns in the data, while also highlighting diminishing returns from increased model complexity under constrained data and computational settings.

4.2 Contextualize with Literature

The results obtained in this study are mostly consistent with prior work on the HIGGS dataset and related high-energy physics classification benchmarks. It also highlights how practical constraints affect the relative performance of different model families. In their study, Baldi et al. Demonstrated that deep neural networks significantly outperform shallow classifiers when trained on the full HIGGS dataset of approximately 11 million samples which achieved a ROC-AUC close to 0.88 [3]. This result established deep learning as a powerful approach for modelling the complex nonlinear feature interactions present in high-energy physics data.

In contrast, the best-performing model in our experiments, which is the feed-forward neural network, achieves a ROC-AUC of approximately 0.83, with gradient-boosted decision trees reaching nearly identical performance with 0.82. While these absolute values are lower than those reported by Baldi et al., the observed gap is expected and can be explained by several methodological differences. Most importantly, our models were trained on a randomly sampled subset of 500,000 events rather than the full dataset. Prior work has shown that the performance advantages of deep neural networks on the HIGGS dataset scale strongly with data volume, as larger datasets allow deeper models to learn subtle high-order feature interactions more effectively [3].

Under reduced data samples, ensemble methods such as gradient-boosted decision trees are known to remain highly competitive. Friedman's original formulation of gradient boosting emphasizes its ability to approximate complex functions through an additive combination of weak learners with strong regularization properties [6]. More recent implementations, such as XGBoost library that we used [7], improve robustness and generalization by using shrinkage, tree depth control, and early stopping [17]. These characteristics help explain why gradient-boosted trees achieve performance comparable to the neural network in our study despite having lower theoretical representational capacity and lower complexity.

Our findings therefore align with existing literature by reinforcing the view that model performance is determined not only by expressiveness, but also by data scale, inductive bias, and training feasibility [2]. While deep neural networks remain the most powerful approach in large-data settings, gradient-boosted trees offer a strong and often more computationally efficient alternative for structured tabular data when resources are constrained. This study contributes additional empirical evidence supporting this trade-off and emphasizes the importance of selecting models based on both theoretical considerations and practical limitations in applied machine learning settings.

4.3 Limitation and Future Work

Several limitations of this study should be acknowledged. The most significant limitation is the reduced dataset size relative to the original HIGGS benchmark. Although the sampled subset preserves class balance and statistical representativeness, it limits the extent to which deep neural networks can fully exploit their expressive capacity. Deep architectures are known to benefit substantially from larger training sets, and training on the full dataset would likely improve absolute performance, particularly for the feed-forward neural network [1].

A second limitation arises from computational constraints. All experiments were conducted on a single CPU-based machine, which restricted both the scale of hyperparameter searches and the complexity of neural network architectures that could be evaluated. As a result, architectural exploration for the multilayer perceptron was deliberately conservative and more computationally intensive optimization strategies were not pursued. While early stopping was effective in preventing overfitting, more advanced regularization techniques or larger models could potentially yield further gains under less restrictive hardware conditions [17].

In addition, the study relies exclusively on high-level engineered features provided by the HIGGS dataset. While these features are physics-informed and widely used, they limit exploration of end-to-end representation learning from lower-level detector information. Furthermore, evaluation focused primarily on ROC-AUC, which measures ranking performance but does not directly account for physics-driven cost functions or operating points relevant to experimental analyses.

Several directions for future work naturally follow from these limitations. First, threshold optimization based on physics-motivated cost functions could be used to better reflect real experimental priorities such as maximizing signal significance under background constraints [18]. Second, deeper neural architectures or alternative activation functions could be evaluated when sufficient computational resources are available, allowing a more thorough investigation of model capacity effects [17]. Finally, feature importance and interpretability analyses could provide valuable insight into which physical variables contribute most strongly to classification performance, improving and supporting downstream physics analyses mainly for gradient-boosted trees [16][19].

4.4 Updates based on the poster presentation feedback

After the poster presentation, several parts and aspects of the project were updated based on the feedback received during the presentation session. First, the initial version of the poster placed relatively strong emphasis on particle physics background. To overcome this issue, we shifted the focus toward the machine learning formulation of the problem, using the physics context primarily to motivate the dataset and task rather than as a central topic in the report and the documentation.

Second, additional details were added to clarify the dataset composition and sampling strategy. The report now explicitly discusses the original scale of the HIGGS dataset, the preservation of label balance during random subsampling, and the rationale for reducing the dataset size to 500,000 samples for computational feasibility.

Third, the description of captures the theoretical foundations of the models was expanded. Each model is now motivated from a machine learning perspective, with clearer explanations of their

inductive biases and appropriate references to foundational literature.
Finally, the discussion section was strengthened by providing a more detailed comparison between our results and those reported in prior work, as well as a clearer articulation of limitations and future research directions. These updates collectively improve the technical depth, transparency, and coherence of the report while directly addressing the feedback received during the poster session.

5 Conclusion

In this project, we studied the problem of distinguishing signal from background events in high-energy physics using the HIGGS dataset which is a large-scale and widely used benchmark for binary classification problems. We were motivated by the limitations of traditional statistical approaches in high-dimensional settings, and we implemented three machine learning model families: logistic regression as a linear baseline, gradient-boosted decision trees as a nonlinear ensemble method, and a feed-forward neural network as a deep learning approach. We evaluated model performance by using ROC-AUC score and analysed how increasing model complexity affects classification performance under realistic computational constraints by using a consistent training, validation, and evaluation pipeline.
Our results show that nonlinear models substantially outperform the linear baseline which confirms the importance of modelling nonlinear feature interactions in this domain. Gradient-boosted trees and the neural network achieve comparable performance on the reduced dataset, with only a marginal advantage for the neural network. Moreover, these results highlight diminishing returns from increased model capacity when data scale and resources are limited. Beyond performance metrics, this study provided valuable insight into the trade-offs between expressiveness, inductive bias, and practical feasibility in applied machine learning. This analysis could be extended this by training on the full dataset with accelerated hardware, exploring physics-driven decision thresholds, and incorporating interpretability analyses to better connect machine learning predictions with experimental objectives in high-energy physics.

Contributions

Tolga Kuntman

- Adapted and extended the logistic regression implementation provided in the lab assignments.
- Trained, tuned, and evaluated the logistic regression model, including performance analysis and interpretation.
- Designed, implemented, trained, and evaluated the feed-forward neural network models.
- Conducted analysis of the reference literature, including the original HIGGS dataset paper and related machine learning studies.
- Contributed to the interpretation of results and the writing of the discussion and conclusion sections.
- **Michel Moussally**
- Designed and implemented the data sampling strategy, including reducing the dataset from 11 million to 500,000 samples while preserving label balance.

- Performed feature preprocessing, feature engineering normalization for GDBT and MLP.
- Implemented, trained, and evaluated the gradient-boosted decision tree models.
- Contributed to experimental design decisions and result validation.

References/Bibliography

[1] CERN, “Higgs boson machine-learning challenge,” May 19, 2014. [Online]. Available: https://home.cern/news/news/computing/higgs-boson-machine-learning-challenge

[2] C. Adam-Bourdais, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau, “The Higgs boson machine learning challenge,” in Proc. NIPS 2014 Workshop on High-Energy Physics and Machine Learning, Montreal, Canada, Dec. 2014, pp. 19–55.

[3] P. Baldi, P. Sadowski, and D. Whiteson, “Searching for exotic particles in high-energy physics with deep learning,” Nature Communications, vol. 5, no. 1, Jul. 2014, Art. no. 4308, Doi: 10.1038/ncomms5308.

[4] D. Whiteson. "HIGGS," UCI Machine Learning Repository, 2014. [Online]. Available: https://doi.org/10.24432/C5V312.

[5] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[6] J. H. Friedman, “Greedy function approximation: A gradient boosting machine.,” Ann. Statist., vol. 29, no. 5, Oct. 2001, Doi: 10.1214/aos/1013203451.

[7] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Belmont, CA, USA: Wadsworth, 1984.

[8] T. Chen et al., “XGBoost: Extreme Gradient Boosting,” software library, 2015. [Online]. Available: https://github.com/dmlc/xgboost

[9] M. Tomasini, “An Introduction to Multilayer Networks,” 3966, 2015, Doi: 10.13140/RG.2.2.16830.18243.

[10] V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines, in Proc. 27th Int. Conf. Machine Learning (ICML), 2010, pp. 807–814.

[11] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed. New York, NY, USA: Springer, 2009.

[12] L. Bottou, “Stochastic gradient descent tricks,” in Neural Networks: Tricks of the Trade, 2nd ed., G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Germany: Springer, 2012, pp. 421–436, doi: 10.1007/978-3-642-35289-8_25.

[13] L. Prechelt, “Early stopping—But when?” in Lecture Notes in Computer Science, vol. 1524, Berlin, Germany: Springer, 2000, doi: 10.1007/3-540-49430-8_3.

[14] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” software library, 2016. [Online]. Available: https://pytorch.org/

[15] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.

[16] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” Proc. 22nd ACM SIGKDD, Int. Conf. Knowledge Discovery and Data Mining, pp. 785–794, 2016.

[17] J. Heaton, “Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep Learning. Cambridge, MA, USA: MIT Press, 2016, 800 pp,” Genetic Programming and Evolvable Machines, vol. 19, Oct. 2017, doi: 10.1007/s10710-017-9314-z.

[18] G. Cowan, K. Cranmer, E. Gross, and O. Vitells, “Asymptotic formulae for likelihood-based tests of new physics (vol. 71, p. 1554, 2011),” Eur. Phys. J. C, vol. 73, Jul. 2013, doi: 10.1140/epjc/s10052-013-2501-z.

[19] S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” 2017, arXiv. doi: 10.48550/ARXIV.1705.07874.

Formatting (Delete this section in your submission)

Feel free to adjust the specific sections according to your needs (e.g. combine introduction and related work or separate the experiments from the discussion. You need to use one -column layouts. The paper size is standard A4 or 8.5 x 11 inches. Your font size must be no smaller than 11pt. Do not use less than 0.5-inch margins. You are not required to type your report in latex. If you use latex (or even Microsoft Word), we highly recommend using a conference/journal template (e.g. NIPS, IEEE, ICML). They generally provide both .tex and .doc templates. When you submit your final report, it must be in PDF format. Please save your proposal file in pdf and name as: Report_ [coach name] _[C/R] _[surname1] _[surname2].pdf

Where:

- coach name = your assigned coach number
- C/R = task type (C = classification, R = regression)
- surname1_surname2 = the last names of both team members

Examples:

- Coach Diwas, Regression, team Janssen & De Smet → Report_Diwas_R_Janssen_De Smet.pdf
- Coach Meixing, Classification, team Li & Garcia → Report_Meixing_C_Li_Garcia.pdf