Advanced Laboratory Course: Particle Physics

Data Analysis with IceCube Monte Carlo Simulation Data

Edgar Eduardo Mata Mendoza
matamd2018@licifug.ugto.mx

Josue Salvador Elizalde Palacios
jojosu.uni@gmail.com

Technical University of Dortmund
June 2024

Contents

1 Introduction

IceCube is an experiment for detection of high-energy neutrinos and muons and is designed to study the cosmos from this particles. It is located deep under the South Pole ice. This observatory, which includes the IceTop and DeepCore arrays, investigates violent astrophysical phenomena such as stellar explosions and gamma-ray bursts. The IceCube Collaboration addresses fundamental questions in physics, such as the nature of dark matter and the properties of neutrinos.

Neutrinos are not directly observed, but when they interact with ice, they produce electrically charged secondary particles that emit Cherenkov light when they travel through ice faster than light does in this medium. IceCube sensors detect light that can be used to reveal the direction and energy of muons and neutrinos. However, neutrino signals are often obscured by more frequent background events from atmospheric muons. In this lab report, data simulated for the neutrino telescope IceCube is analyzed. The objective of this analysis is to use machine learning to classify IceCube events as either signal events (neutrinos) or background events (atmospheric muons).

Preparation, attribute selection and various multivariate machine learning models are applied to classify the data. Several algorithms including Naive Bayes (NB), Random Forest (RF), and k-Nearest Neighbors (KNN), are explored to determine which best identifies neutrino signals and rejects background noise. The performance of these models is then evaluated using precision, recall, the area under the Receiver Operating Characteristic (ROC) curve and other parameters to understand the effectiveness of the classifiers.

2 Overview of IceCube Detector

The IceCube Neutrino Observatory consists of a cubic-kilometer detector situated in ice at a depth between 1450 m and 2450 m and a square-kilometer detector array at the surface. The detector employs optical sensors to detect light of charged particles generated by neutrinos in the ice or Earth's crust. In addition, IceCube also includes a more densely instrumented part called DeepCore and an extensive air shower array on the surface called IceTop [Aea12]. A schematic representation of the IceCube Neutrino Observatory is shown in Figure 1.

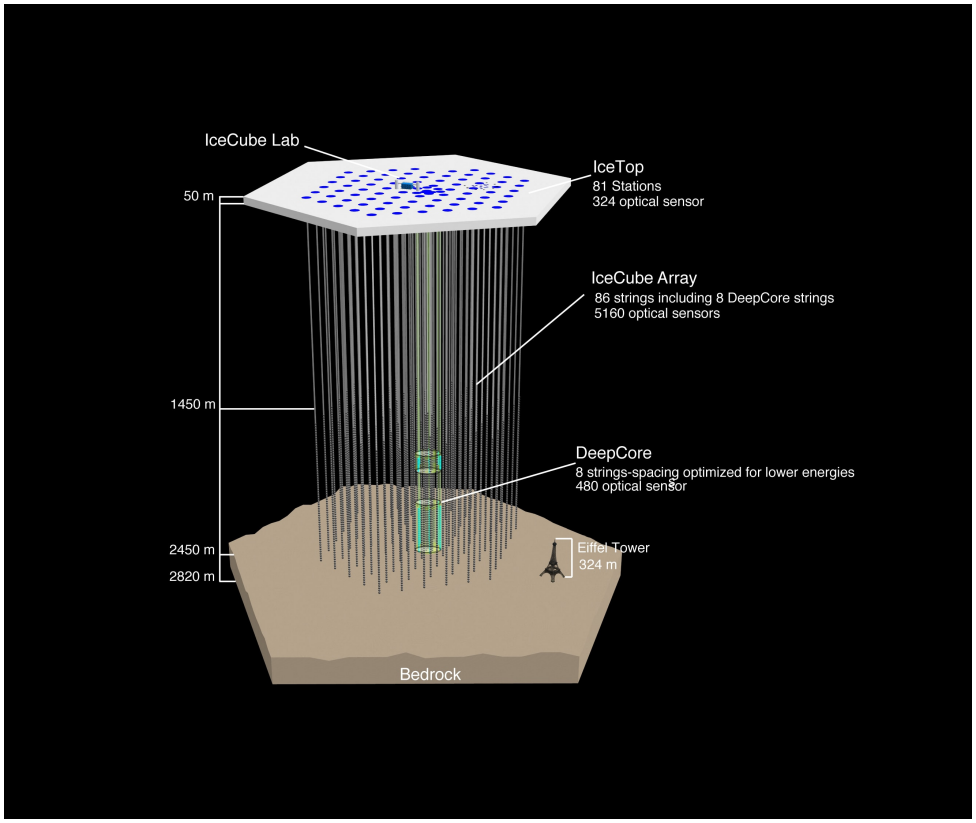


Figure 1: The IceCube Neutrino Observatory. Credit: IceCube Collaboration.

IceTop, consists of 162 tanks of ice, each instrumented with two standard IceCube sensors, to detect showers of secondary particles generated by interactions of high-energy cosmic rays in the atmosphere.

The in-ice component of IceCube consists of $5,160$ digital optical modules (DOMs), each equipped with a ten-inch photomultiplier tube and electronics. The DOMs are attached to vertical strings, frozen into 86 boreholes, and arrayed over a cubic kilometer from $1,450$ to $2,450$ meters depth. The strings, deployed on a hexagonal grid with 125 meters spacing, each hold 60 DOMs with a vertical separation of 17 meters.

At the DeepCore subdetector, eight strings are more compactly arranged, with a horizontal separation of about 70 meters and a vertical DOM spacing of 7 meters. This configuration lowers the neutrino energy threshold to about 10 GeV, enabling the study

of neutrino oscillations.

3 Analysis strategy

To extract meaningful physics results from the IceCube data, a robust analysis strategy is essential. This involves prepossessing the raw data, applying selection criteria and compare different machine learning algorithms to improve classification accuracy.

3.1 Attribute selection

Attribute (features) selection is a crucial step in the process of building machine learning models. It involves selecting the most relevant features from the dataset that contribute the most to predicting the target variable. The goal is to improve computational cost.

In this analysis the Minimum Redundancy Maximum Relevance (mRMR) selection was used. It does not rely on a specific learning algorithm but instead considers the probabilities of available variables. It utilizes joint information between variables x and y, defined as [Ast24]:

$$I(x,y) = \int p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dx\, dy,, \tag{1}$$

where $p(x)$, $p(y)$ and $p(x,y)$ are probability density functions. This method iteratively selects variables that have strong correlations with the target variable while minimizing correlations with each other.

3.2 Quality Parameters

To assess the performance of a classification model, several key parameters were used based on correctly and incorrectly classified events as signal (true positive, tp) or background (true negative, tn), and misclassifications as false positive (fp) or false negative (fn) [Ast24]. These parameters include:

- **Accuracy (A)**: The ratio of correctly classified events to the total number of events.
- **Precision (p)**: The ratio of true positive events to the sum of true positive and false positive events:
$$p = \frac{tp}{tp + fp} \tag{2}$$
- **Recall (r)**: The ratio of true positive events to the sum of true positive and false negative events:
$$r = \frac{tp}{tp + fn} \tag{3}$$
- **f_β-score**: This score with $\beta = 0.1$ was used to choose the best value for the threshold of the model:
$$\mathrm{f}_\beta = \frac{(1+\beta^2)\cdot p \cdot r}{\beta^2 \cdot p + r} \tag{4}$$

- **ROC Curve**: The ROC curve plots the True Positive Rate ($\mathrm{TPR}(\tau_c)$) against the False Positive Rate ($\mathrm{FPR}(\tau_c)$) at various threshold settings τ_c.

$$\mathrm{TPR}(\tau_c) = \frac{tp(\tau_c)}{tp(\tau_c) + fn(\tau_c)} \tag{5}$$

$$\mathrm{FPR}(\tau_c) = \frac{fp(\tau_c)}{fp(\tau_c) + tn(\tau_c)} \tag{6}$$

  The area under the curve (A_{ROC}) provides a value to compare the performance of different models, a value equal to 1 for a perfect classification model and 0.5 if the classes are randomly guessed. Values < 0.5 indicate confusion between both classes.

3.3 Cross-Validation

Cross-validation was used to estimate the error associated with the quality parameters mentioned above. In cross-validation, the training dataset is divided into n parts. The learner is trained on $n-1$ parts, and the model created is then evaluated on the remaining part. This process is repeated n times so that each part serves as the test dataset exactly once. As a result, n values are obtained for the quality parameters mentioned earlier, allowing for the calculation of the mean error associated with these parameters [Ast24].

4 Analysis

The main objective of this analysis is to evaluate and compare the performance of three classifiers (NB, RF, KNN) in distinguishing true neutrino events (signal, labeled as 1) from unrelated data (background, labeled as 0), in Monte Carlo simulated datasets from the IceCube experiment. The dataset includes three distinct subsets: the first two datasets serve as signal and background for training, while the third dataset functions as a test set for predicting labels.

4.1 Preparing the data

The primarily task of this section was to prepare the provided datasets of signal and background for the analysis. After loading the datasets and had a first look at the data, the first step consisted in remove any attributes that appeared only in one of the two datasets. Then, columns corresponding to Monte Carlo truths, event ids and weights for the training process were removed. Rows with `NaN` or `Inf` values were also removed. Finally, the background and signal training datasets were combined and separated into the features and target (labels) training datasets. The final feature trining dataset consisted in a shape of 36 000 rows and 187 columns (attributes).

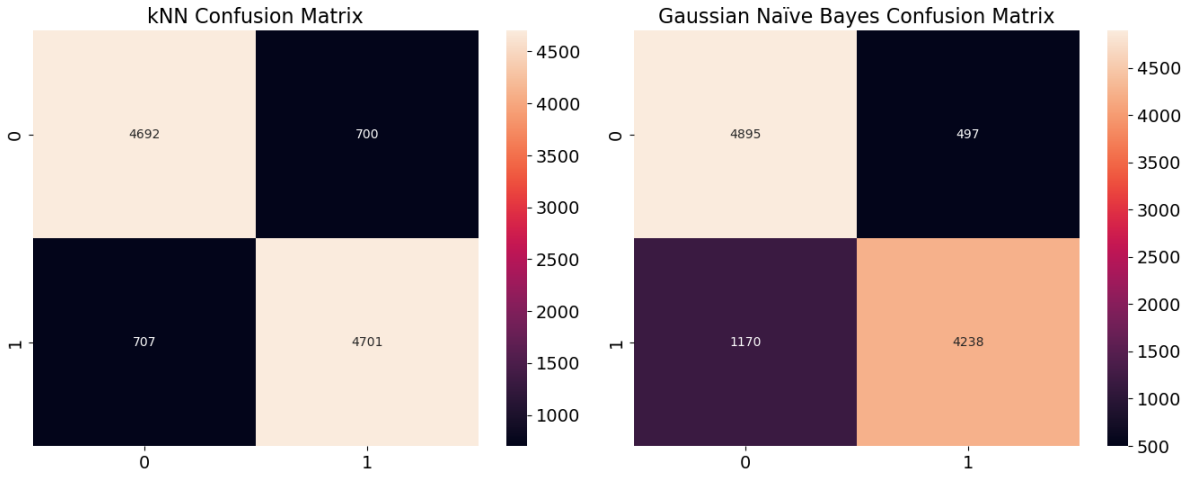
4.2 Attribute Selection

After completing the data preparation, attribute selection was carried out using the mRMR method to reduce redundancy and computation time while maintaining predictive accuracy. A decision was made to maintain 12 attributes, resulting in the following selection:

1. `LineFit_TTParams.lf_vel_z`
2. `SplineMPEDirectHitsC.n_dir_strings`
3. `HitStatisticsValues.z_travel`
4. `SplineMPEFitParams.rlogl`
5. `SplineMPEDirectHitsA.dir_track_length`
6. `LineFit_TT.zenith`
7. `SplineMPEDirectHitsA.n_dir_strings`
8. `MuEXAngular4.zenith`
9. `MuEXAngular4_Sigma.value`
10. `MPEFitHighNoise.zenith`
11. `SplineMPEDirectHitsA.n_dir_doms`
12. `SplineMPE.zenith`

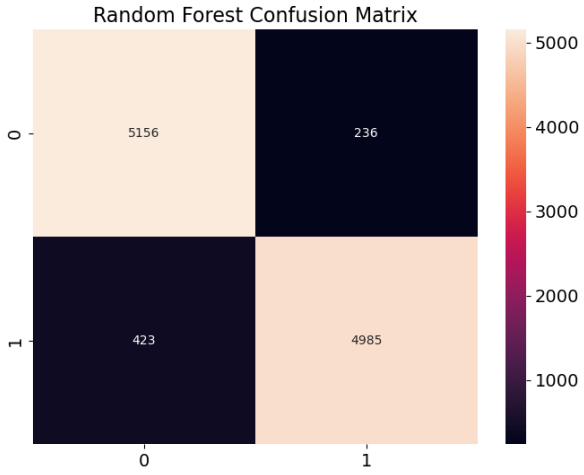
4.3 Multivariate Separation

In this section, the k-Nearest Neighbors, Naive Bayes, and Random Forest classifiers were employed for event classification. Confusion matrices were generated for each learner to evaluate their effectiveness in separating events. The results are shown in Figure 2.



(a) k -Nearest Neighbors.

(b) Naive Bayes.



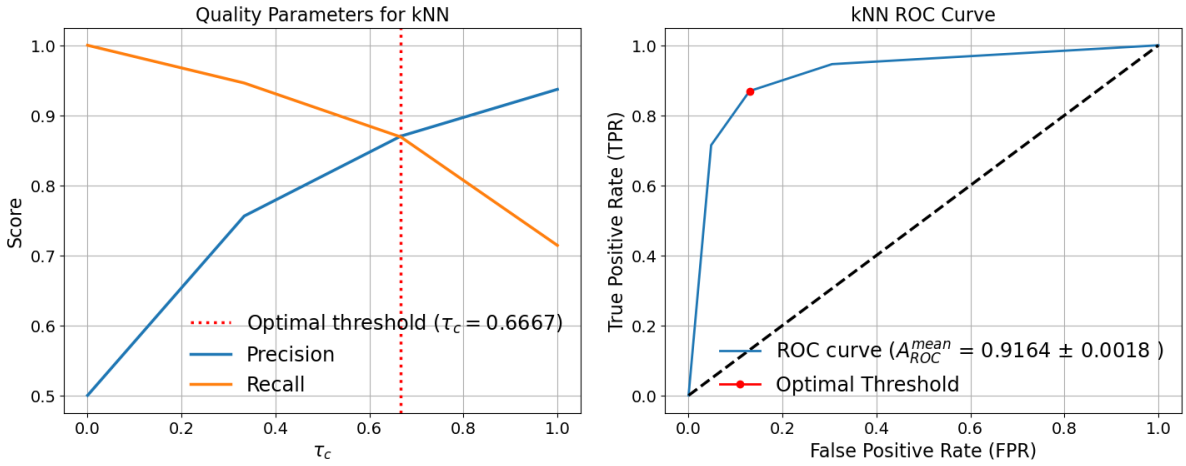
(c) Random Forest.

Figure 2: Confusion Matrix for the different classifiers.

Precision and recall for different values of τ_c, as well as ROC curves and A_{ROC} were calculated for the three learners. Accuracy and f_β scores were also computed to determine the optimal τ_c for each learner. Cross-validation was performed to derive the mean and the mean error of all quality parameters mentioned above. Optimal threshold in ROC curve is based on f_β-score with $\beta = 0.1$. The results are presented in Table 1 and the corresponding figures for precision, recall, ROC curves, and A_{ROC} for each learner are shown in Figures 3, 4 and 5.

Model	Average Accuracy	Average Precision	Average Recall	Average A_{roc}	Average f_β
RF	0.9360 ± 0.0015	0.9521 ± 0.0020	0.9175 ± 0.0024	0.9810 ± 0.0009	0.9517 ± 0.0020
NB	0.8433 ± 0.0018	0.8900 ± 0.0041	0.7816 ± 0.0034	0.9248 ± 0.0016	0.8888 ± 0.0041
kNN	0.8696 ± 0.0027	0.8681 ± 0.0048	0.8700 ± 0.0023	0.9164 ± 0.0018	0.8681 ± 0.0047

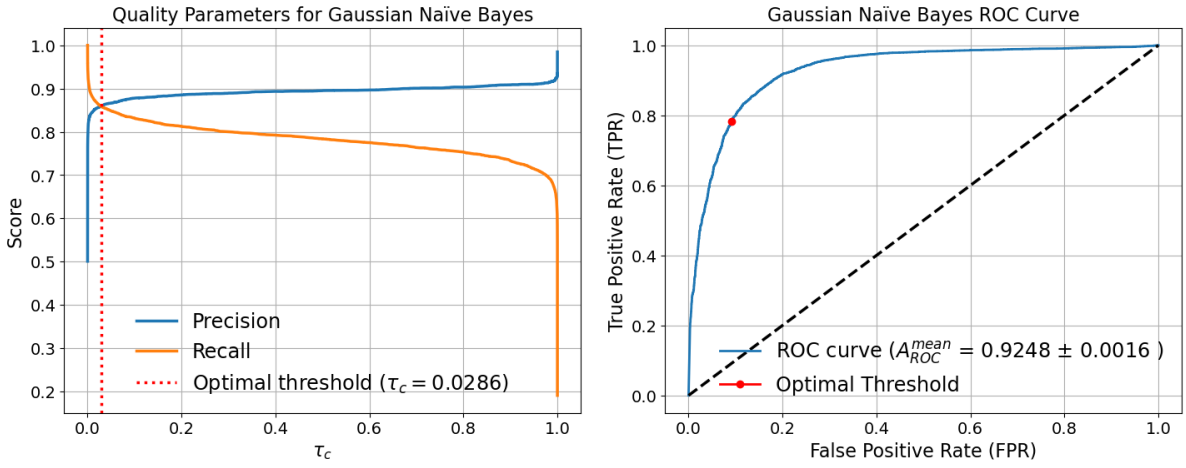
Table 1: Mean values and corresponding errors for quality parameters of different machine learning models after 10-fold Cross-Validation.



(a) Quality parameters.

(b) ROC curve.

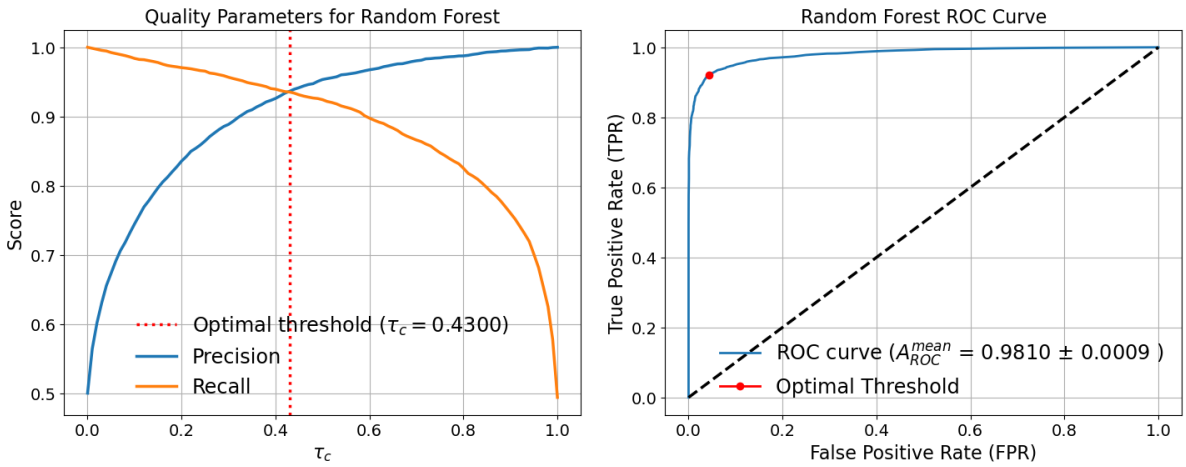
Figure 3: Quality Parameters for different threshold values and ROC curve for the k Nearest Neighbors.



(a) Quality parameters.

(b) ROC curve.

Figure 4: Quality Parameters for different threshold values and ROC curve for the Naïve Bayes.



(a) Quality parameters.

(b) ROC curve.

Figure 5: Quality Parameters for different threshold values and ROC curve for the Random Forest.

5 Conclusion

Accuracy and f_β scores were calculated to find the optimal τ_C for each classifier, with the Random Forest showing the highest accuracy. According to the A_{ROC} values, all classifiers performed well, each scoring above 0.9000, indicating strong classification power. The Random Forest achieved the highest A_{ROC} value of 0.9810 ± 0.0009. It also maintained a better balance between precision and recall across different values compared to Naive Bayes and KNN, further supporting its effectiveness. Additionally, it demonstrated the best performance metrics with the lowest error margins across the 10-fold cross-validation, indicating superior performance in distinguishing between signal and background events.

Taking this into account and after evaluating the remaining quality parameter scores, it was concluded that the Random Forest classifier demonstrated the best performance. Consequently, this model, along with its corresponding threshold, was employed to make predictions on the provided test dataset. The results from these predictions are presented in the accompanying .csv file.

References

[Aea12] R. Abbasi and Y. Abdou et al. The design and performance of icecube deepcore. *Astroparticle Physics*, 35(10):615–624, 2012.

[Ast24] Astroparticle Physics, TU Dortmund University. Advanced laboratory course manual: Particle physics data analysis with icecube monte carlo simulation data, April 2024.