Data Analysis with IceCube Monte Carlo Simulation Data

Lukas Bertsch
lukas.bertsch@tu-dortmund.de

Tabea Hacheney
tabea.hacheney@tu-dortmund.de

Tom Troska
tom.troska@tu-dortmund.de

Start of course: 28 June 2024

TU Dortmund University – Faculty of Physics

Contents

1 Motivation

The study of particles from astrophysical sources is of great interest to understand stellar and galactic processes, as well as our universe itself. Neutrinos are well suited for these studies, since they are uncharged and only interact weakly, allowing to trace measured neutrinos on earth back to their origin in far away astrophysical sources.
In this analysis, a selection of neutrino events is performed using Monte Carlo simulation data from the IceCube experiment. A minimum redundancy, maximum relevance ($mRMR$) selection is employed to determine the most suitable features for a multivariate analysis separating signal and background events. Three different machine learning algorithms are compared and their performance on the classification task is evaluated.

2 Theory

In this section, the theory aspects of the analysis are explained. At first, the basics of astroparticle physics and neutrino detection are described and then, machine learning methods utilized in this study are discussed.

2.1 Fundamentals of Astroparticle Physics and Cosmic Rays

The earth is constantly hit by ionizing, high energy particles originating from astrophysical sources in the universe which are called *cosmic rays*. The majority of these particles are protons, light and heavy nuclei and electrons that interact with the earth's atmosphere and cause large cascades of particle decays ('*air showers*') which can be measured as cosmic radiation on earth. Since these particles are charged, they are deflected by galactic and extragalactic magnetic fields on their way to earth and cannot be traced back to their origin. The energy spectrum of these charged particles extends up to 10^{20} MeV and has a flux described by the power law

$$\frac{\mathrm{d}\Phi}{\mathrm{d}E} = \Phi_0 E^{\gamma}, \tag{1}$$

where $\gamma \approx -2.7$ is the spectral index. Apart from the charged particles, high energy gamma rays and neutrinos from outer space reach the earth. Since these particles are not charged, they can in theory be traced back to their origin. Neutrinos only interact via the weak interaction and have very small cross sections, allowing them to penetrate dense regions in the universe on their way to earth which is not possible for gamma rays that interact via the electromagnetic interaction. This analysis will focus on neutrinos that are measured by the IceCube experiment, which is described in detail in section 3. At IceCube, atmospheric and cosmics neutrinos are measured. The atmospheric neutrinos are created in the previously mentioned air showers and are further categorized into conventional and prompt neutrinos. Conventional neutrinos originate from kaon and pion decays to muons and muon antineutrinos. Since these particles have a comparatively long lifetime, they lose a significant amount of energy before they decay, resulting in an energy spectrum with a spectral index of $\gamma \approx -3.7$. Prompt neutrinos stem from

semi-leptonic decays of heavy hadrons like D mesons and Λ baryons which have a short lifetime and therefore do not lose as much energy, resulting in a energy spectrum with $\gamma \approx -2.7$, similar to the spectrum of charged cosmic rays. Here, the desired signal to be measured is the cosmic neutrinos from astrophysical sources. Under the assumption of shock acceleration [1], the flux of these neutrinos has a spectral index of $\gamma \approx -2$.

2.1.1 Measurement of neutrinos with the IceCube detector

In IceCube, neutrinos are measured via *Cherenkov light* of secondary particles that are created in interactions with the ice molecules. Cherenkov light is emitted when a charged particle traverses a medium with a higher velocity than the respective speed of light in the medium. The speed of light in a medium with refractive index n is given by $c = {}^{c_0}/_{n}$, where c_0 is the speed of light in vacuum. The neutrinos can interact via the charged current (CC)

$$\nu_l(\bar{\nu}_l) + A \rightarrow l^{\mp} + X \tag{2}$$

and the neutral current (NC)

$$\nu_l + A \rightarrow \nu_l + X, \tag{3}$$

where A are the ice nuclei and X all other finals state particles in the reaction. Different signatures of the events allow to distinguish between different lepton flavors in Equation 2 and NC events. Electrons and NC events create circular, cascade like events, whereas muons create long tracks in the detector. Tau leptons have a short lifetime and therefore cause events with two nearby circular cascades. Further information on the IceCube detector system is given in section 3.

2.2 Feature Selection and Multivariate Analysis

In this section, the feature selection and machine learning algorithms that are employed in this analysis are briefly described.

2.2.1 mRMR Selection

In a mRMR (minimum redundancy, maximum relevance) selection, a set of variables (features) is iteratively selected that strongly correlates with the target (signal or background) and has a low redundancy (correlation between features). For this purpose, the joint information of two variables x, y

$$I(x,y) = \int p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) \mathrm{d}x\mathrm{d}y \tag{4}$$

is considered, where $p(x/y)$ are the respective probability functions. Here, the mRMR implementation in the python package *mrmr-selection* [2] is used.

2.3 Cross-validation

With the help of the cross-validation methods, quality parameters for a classifier can be determined. This is done by splitting the dataset into n subsets. The classifier is then trained on $n-1$ subsets with the remaining subset being used to calculate the quality parameters. By perfoming n iterations, so that each subset is once used as the test dataset, a total of n values for each quality parameter is gathered. The mean and standard deviation can then be calculated.

2.3.1 Naïve Bayes Classifier

In a naïve bayes classifier, Bayes theorem on conditional probabilites

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{5}$$

is used to express the likelihood of an event belonging to a class $A(\overline{A})$ (signal (background)) using features B_i. With n attributes, the measure

$$Q = \prod_{i=1}^{n} \frac{p(B_i|A)}{p(B_i|\overline{A})} \tag{6}$$

is used to distinguish between signal with $Q > 1$ and background.

2.3.2 Decision Trees

Decision trees are binary classifiers, that separate between different classes by subsequently applying binary cuts on the available variables. For each decision point (*node*), the cut that maximizes the separation between the classes is searched. The data is divided into two subsets after each cut, till a maximum number of cuts (depth) is reached or the classes are fully separated. To minimize overtraining, different techniques can be employed to make the classification more robust. For example, a random forest can be used. In a random forest, an ensemble of individual decision trees is trained, where each tree is trained on a subset of the training data using k randomly selected variables. To get a classification, the arithmetic mean of all individual tree decisions is taken.

2.3.3 k-Nearest Neighbours Classifier

A k-Nearest Neighbours (kNN) classifier is a so-called lazy learner, because the kNN algorithm has no training phase. The training data is saved in the classifier and a classification is performed by computing the euclidean distance of the event to classify to all previously saved datapoints. The most frequent label (class) among the k nearest neighbours (i.e. the saved training data) is then assigned to the event.

2.3.4 Evaluation Metrics

In order to evaluate the performance of a classifier on the classification task, different metrics are used. The classification output can be grouped into four categories: Correctly classified signal events (true positives '*tp*'), correctly classified background (true negatives '*tn*'), background events falsely classified as signal (false positives '*fp*') and signal events falsely classified as background (false negatives '*fn*'). With these definitions, the accuracy

$$a = \frac{\mathrm{tp} + \mathrm{tn}}{\mathrm{tp} + \mathrm{tn} + \mathrm{fn} + \mathrm{fp}}, \tag{7}$$

the precision

$$p = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}}, \tag{8}$$

and the recall

$$r = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}} \tag{9}$$

can be defined to evaluate the performance of the classification. Another useful metric is the f_β score

$$f_\beta = (1 + \beta^2)\frac{p \cdot r}{\beta^2 p + r}, \tag{10}$$

which is the harmonic mean of precision and recall with the recall weighted by a factor β. These metrics depend on the classification threshold t that is applied to separate signal and background by a cut on the classifiers output, which is typically a value between 0 and 1, where 1 indicates a high probability of being signal and vice versa. A metric independent of the threshold t can be obtained by the *Receiver Operating Characteristic* (ROC) curve. In the ROC curve, the true positive rate (*TPR*) is plotted against the false positve rate (*FPR*), with the TPR and FPR defined as

$$\mathrm{TPR}(t) = \frac{\mathrm{tp}(t)}{\mathrm{tp}(t) + \mathrm{fn}(t)} \qquad \mathrm{FPR}(t) = \frac{\mathrm{fp}(t)}{\mathrm{fp}(t) + \mathrm{tn}(t)}, \tag{11}$$

while varying the threshold t. The area under the curve (AUC-score) is a measure for the quality of the classification and takes the values 0.5 for random guessing and 1 for a perfect classification. An AUC-score smaller than 0.5 indicates, that the model confuses signal and background, which can easily be corrected by inverting the classification.

3 The IceCube Detector

The experimental data was taken at the IceCube experiment, located at the geographic South Pole at a depth of 1450 m to 2450 m. The detector of IceCube consists of the in-ice array, DeepCore and IceTop and is used to detect high-energy neutrinos and muons. It is formed by 86 cables each connected to 60 Digital Optical Modules (DOMs), consisting of a photomultiplier and a single-board computer, resulting in a total of 5160 photomultipliers detecting weak Cherenkov light of high-energy charged particles.

Seven of those cables are more densely packed with DOMs and form the DeepCore, which is needed to detect particles with lower energies with sufficient efficiency. While the energy threshold at the rest of the detector is approximately 100 GeV, the DeepCore has an energy threshold of just ≈ 10 GeV. This is achieved by the smaller distance and higher efficiency of the used photomultipliers.
The IceTop detector is used to detect air showers, which are needed to study cosmic rays, and it is also used as a veto for detected particles in the in-ice array.

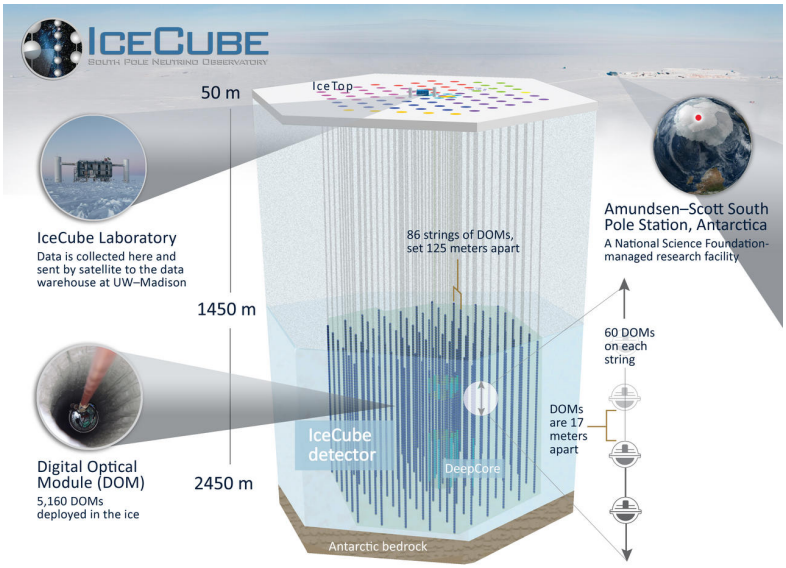


Figure 1: A schematic view of the IceCube detector [3].

4 Analysis Strategy

In this analysis, *starting events* are used to discard atmospheric muon events, since only cosmic neutrinos are of relevance. These *starting events* come from neutrino interactions within the detector and can therefore be used to distinguish between muons coming from the atmosphere and muons coming from interactions with neutrinos. Another strategy is to apply a cut on the reconstructed zenith angle, since atmospheric muons cannot come from below the detector (traveling through earth). Since the reconstruction of the direction of the event is not precise enough, the cut only improves the signal-to-noise ratio up to $1 : 10^3$. For further separation, machine learning methods are used to distinguish between events deriving from astrophysical neutrinos (signal) and those, that have been incorrectly reconstructed.
To train a classifier for signal-background separation, the data has to be properly prepared first. For this, any attributes consisting of mostly `NaN`s or `Inf`s are discarded. A Monte Carlo simulation of signal and background events is used to train the classifier. Therefore, any attributes only appearing in either the simulation or the real recorded data are

discarded as well. Also, unphysical parameters (e.g Monte Carlo truth attributes with names `Weight, MC, Corsika, I3EventHeader`) are not suited for the training process. The `label` attribute of the simulation includes binary values for either signal (`1`) or background (`0`).
The given dataset includes ≈ 100 attributes. An attribute selection is performed, to reduce the dimensionality and computing time for the classification task. For this, the mRMR method explained in 2.2.1 is applied. After extracting the most useful features, a multivariate learner is trained to efficiently separate signal and background. In this analysis, a *naïve Bayes classifier*, a *Random Forest classifier*, as well as a *kNN classifier* are trained and evaluated using the Precision and Recall for different thresholds t. The best threshold is then chosen by using the f_β score. Additionally, ROC curves are plotted and the area under the ROC curve A_{ROC} is calculated to evaluate the classifier performance independently of the threshold.
The final labels of the real data are then predicted by the model with the best achieved performance.

5 Analysis

For this analysis, three different datasets were provided. First, the `signal_train.csv` and the `background_train.csv` are used to train three different classifiers. For these datasets, the true values are known so that a proper training can be performed. The `test.csv` dataset is unlabeled and the best perfoming classifier's task is the classification of these data.

5.1 Data preparation and attribute selection

The data for the training of the classifiers need to be prepared for this analysis. Therefore, certain non-physical features arising from the Monte Carlo simulations need to be removed. Moreover, all features with more than 10 % non-pyhsical `NaN` or `Inf` values are dropped. The remaining rows that contain `NaN` or `Inf` values are then deleted. Finally, the labels with the information whether a measurement is a true signal or true background event is removed from the dataframe.
The selection of important features is a crucial part of a multivariate analysis. Here, an mRMR selection (see section 2.2.1) is performed to find the most relevant features. This is done with the help of the `python` extension `mrmr_selection` [2]. In Figure 2, the best three features are presented, whereas less important features are depicted in Figure 3.

5.2 Multivariate selection

The Naïve Bayes, the Random Forest and the kNN classifiers are chosen for the multivariate selection. For all three classifiers, the ten best features are used and a k-fold with $n = 5$ splits is performed. The classifiers are trained via the cross-validation method (see section 2.3). A value of $\beta = 0.1$ for the f_β score (10) is used throughout this analysis,

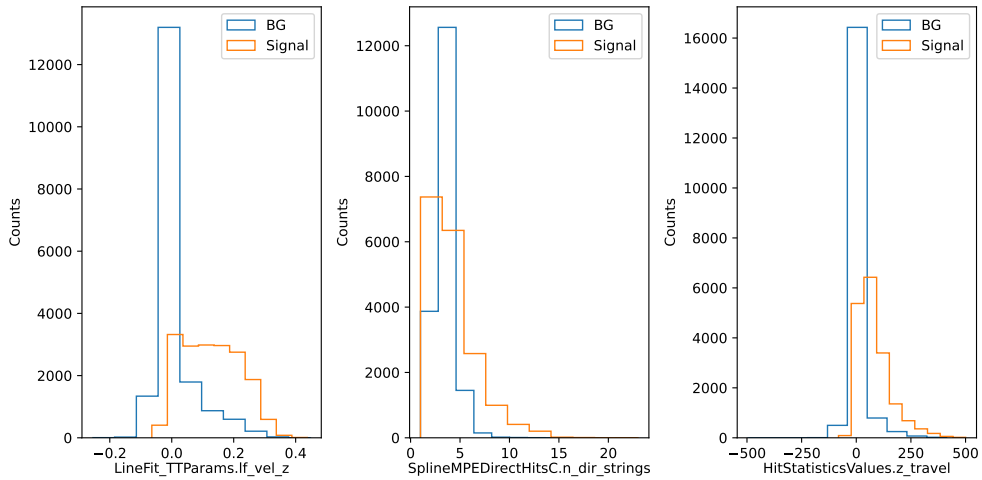


Figure 2: The best three features determined by the mRMR selection.

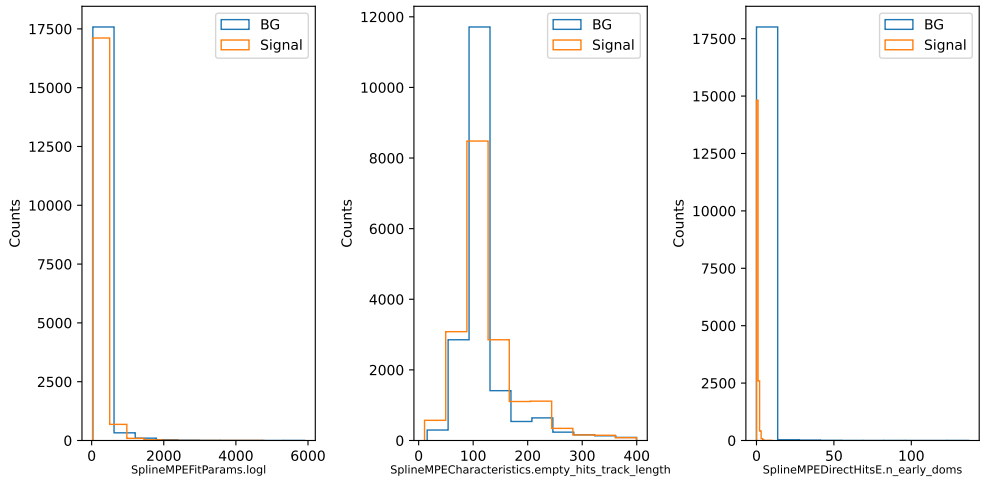


Figure 3: The three worst features determined by an mRMR selection with a total of 100 features.

placing more importance on the precision rather than the recall.
The performance of each classifier is discussed in the following sections.

5.2.1 Naïve Bayes

The working priciple of a Naïve Bayes classifier is explained in section 2.3.1. In Figure 4, the accuracy, precision and recall metrics are portrayed.

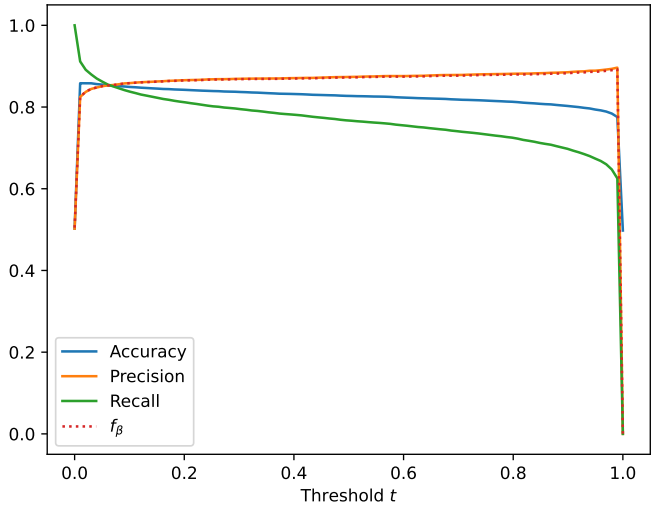


Figure 4: Naïve Bayes: accuracy, precision and recall. Magnification of the relevant part of the f_β score is shown in Figure 5.

The cross-validation of the Naïve Bayes classifier yields the following quality parameters

$$a = 0.827 \pm 0.005$$
$$p = 0.873 \pm 0.006$$
$$A_{\text{ROC}} = 0.9211 \pm 0.0033$$

The best value for the threshold t is determined by finding the maximum of the f_β score. This is depicted in Figure 5 and for the Naïve Bayes classifier, an optimal threshold value of $t_{\text{best}} = 0.9899$ is obtained. The confusion matrix of the classifier with this threshold value is shown in Figure 6.

5.2.2 Random Forest

As the second classifier, a Random Forest is selected. The idea behind decision trees is further addressed in section 2.3.2. Similar to the previous classifier, a cross-validation is

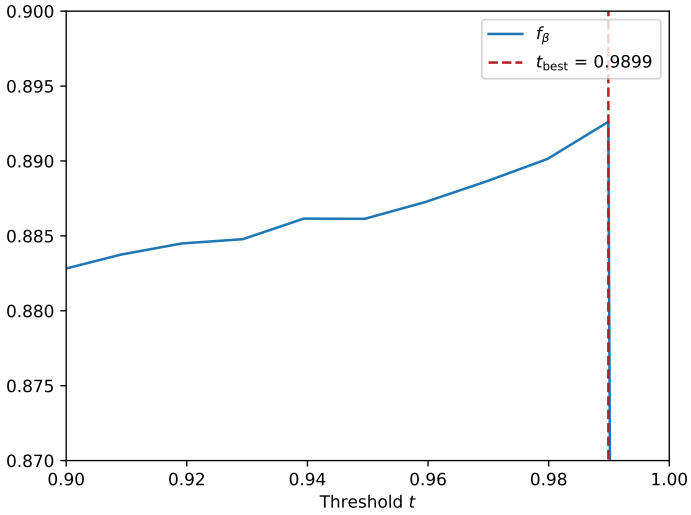


Figure 5: Naïve Bayes: f_β dependend on the threshold t. The maximum value for f_β is marked with dashed lines.

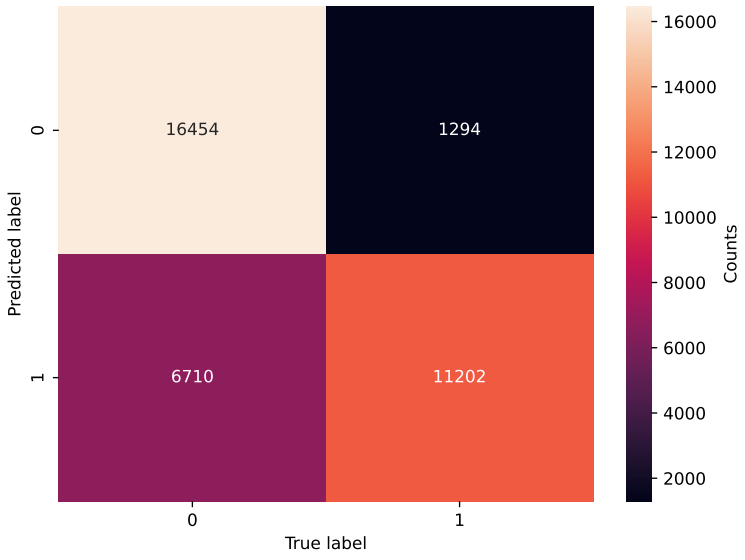
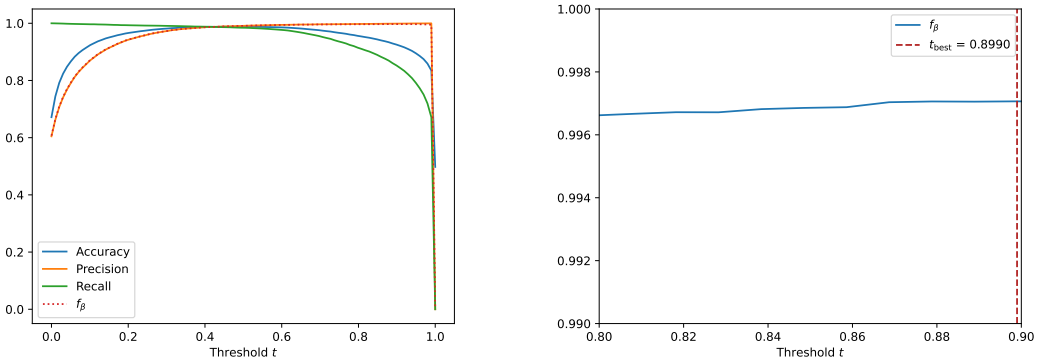


Figure 6: Naïve Bayes: Confusion matrix with threshold value $t_{\text{best}} = 0.9899$.

executed and the quality parameters are returned as

$$\begin{aligned} a &= 0.9403 \pm 0.0019 \\ p &= 0.9544 \pm 0.0022 \\ A_{\mathrm{ROC}} &= 0.9829 \pm 0.0012. \end{aligned}$$

For this classifier, the best threshold value is acquired as $t_{\mathrm{best}} = 0.8889$. In the Figures 7a, 7b and 8, the quality parameters and the confusion matrix are shown.



(a) Accuracy, precision and recall. Magnification of the relevant part of the f_β score is shown in Figure 7b.

(b) The f_β score dependend on the threshold t. The maximum value for f_β is marked with dashed lines.

Figure 7: Qualtity parameters for the Random Forest classifier.

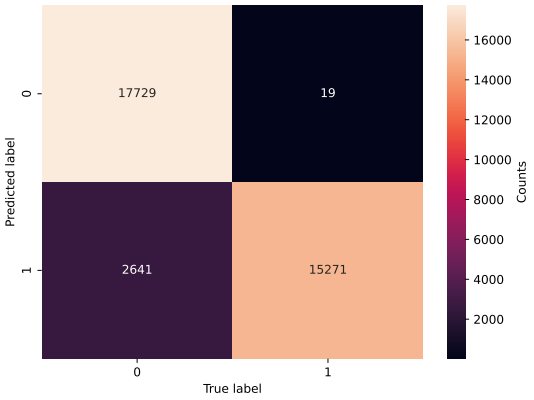
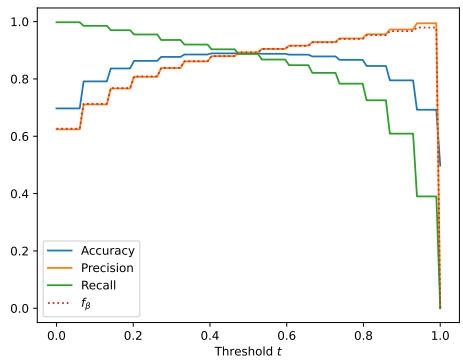


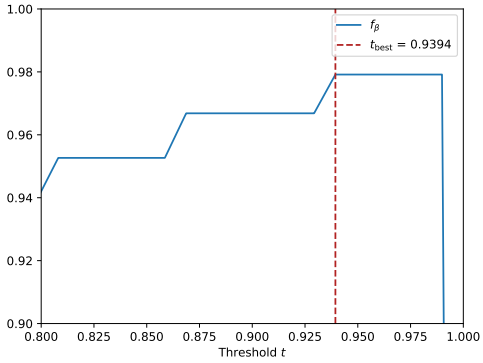
Figure 8: Random Forest: Confusion matrix with threshold value $t_{\mathrm{best}} = 0.8889$.

5.2.3 k nearest neighbors

For the k nearest neighbors (kNN) algorithm, the procedure is the same as before. In section 2.3.3, this classifier is introduced. The Figures 9a, 9b and 10 contain information about the quality parameters and the confusion matrix. The values for accuracy,



(a) Accuracy, precision and recall. Magnification of the relevant part of the f_β score is shown in Figure 9b.



(b) The f_β score dependend on the threshold t. The maximum value for f_β is marked with dashed lines.

Figure 9: Qualtity parameters for the kNN classifier.

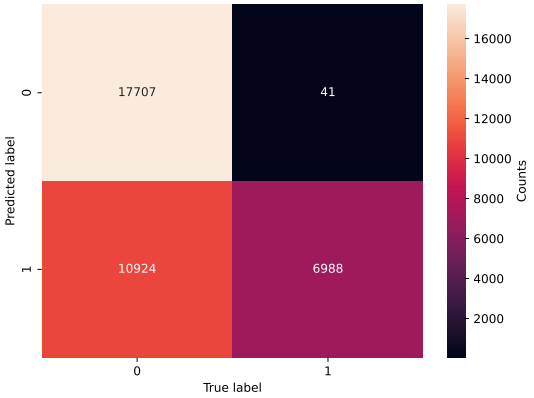


Figure 10: Random Forest: Confusion matrix with threshold value $t_{\text{best}} = 0.9394$.

precision, ROC-AUC score, and f_β score are as follows

$$\begin{aligned} a &= 0.8821 \pm 0.0016 \\ p &= 0.885 \pm 0.007 \\ A_{\text{ROC}} &= 0.9400 \pm 0.0023 \\ f_\beta &= 0.9394. \end{aligned}$$

5.3 Comparison of the learning algorithms

In Figure 11, the ROC curves for all three classifiers are depicted. The values for the true positive (TP) rate and the false positive (FP) rate are extracted from the corresponding confusion matrices. The individual ROC-AUC scores are listed in Table 1.

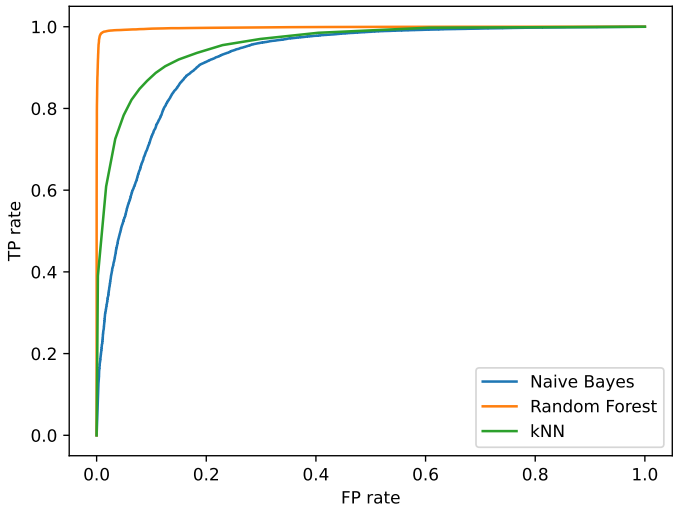


Figure 11: ROC curves for the three different classifiers.

Table 1: ROC-AUC scores for the different classifiers.

Classifier	ROC-AUC score
Naïve Bayes	0.9212
Random Forest	0.9983
kNN	0.9562

The Random Forest classifier is identified as the best-performing model and thus selected to classify the provided `test.csv` dataset. Out of a total of 4000 events, 1633 are classified as signal, while the remaining 2367 are classified as background.

6 Discussion

To classify the 4000 given events as signal or background, a Naïve Bayes classifier, a Random Forest classifier as well as a kNN classifier are trained using the ten most relevant features of the dataset according to an mRMR selection. The performance of the classifiers is evaluated using the area under the ROC curve A_{ROC}, the accuracy and

the precision. The final values of the A_{ROC} for the classifiers are

$$\begin{aligned} A_{\mathrm{ROC,Bayes}} &= 0.9400 \pm 0.0023 \\ A_{\mathrm{ROC,RF}} &= 0.9829 \pm 0.0012 \\ A_{\mathrm{ROC,kNN}} &= 0.9211 \pm 0.0033. \end{aligned}$$

While all three classifiers yield very good results, the Random Forest performs best on the test subset. Therefore, this model is chosen for the final prediction of the labels.
To achieve even higher scores for the quality of the predictions, the number of input features can be optimized and in general more effort can be put into data preparation before training the model. This can be done by scaling the input parameters to achieve a better distinction between the distributions of the used features.
According to the confusion matrices of the classifiers, most misclassified events are background events being classified as signal. This is problematic, since this could lead to an overestimation of signal and therefore potential false discoveries.
Further analysis and optimization for the trained models is needed, to achieve better performance in this classification problem and prevent the many background events from being classified as signal.

References

[1] ENRICO Fermi. 'On the Origin of the Cosmic Radiation'. In: *Phys. Rev.* 75 (8 Apr. 1949), pp. 1169–1174. DOI: 10.1103/PhysRev.75.1169. URL: https://link.aps.org/doi/10.1103/PhysRev.75.1169.

[2] Prasanna Sudan and Nizar Al Khalil. *MRMR-Selection: Minimum Redundancy Maximum Relevance Feature Selection*. Version 0.2.8. 2023. URL: https://pypi.org/project/mrmr-selection/#description.

[3] *IceCube: The world's largest neutrino telescope*. Deutsches Elektronen-Synchrotron DESY. URL: https://astroparticle-physics.desy.de/research/neutrino_astronomy/icecube/index_eng.html (visited on 02/07/2024).