Advanced Lab Course: Particle Physics

IceCube

Leander Flottau
leander.flottau@tu-dortmund.de

Ajeesh Garg
smajgarg@tu-dortmund.de

14. Mai 2024

TU Dortmund University – Department of Physics

Contents

1 Introduction

Astroparticle physics combines both particle physics and astrophysics in a study of elementary particles arriving from the cosmos. This field acquired a radically different nature with the discovery of cosmic rays by Victor Hess in 1912, which introduced a class of high-energy particles that interact with the atmosphere of the Earth. With energies extending up to 10^{20} electronvolts (eV), cosmic rays mainly composed of protons and helium nuclei have been measured on Earth. Their power-law energy spectrum drops very steeply with a spectral index of -2.7.

2 Theory

2.1 Atmospheric Muons and Neutrinos

When cosmic ray particles interact with the earths atmosphere they can produce a number of different secondary particles. Those can decay or interact with the atmsophere

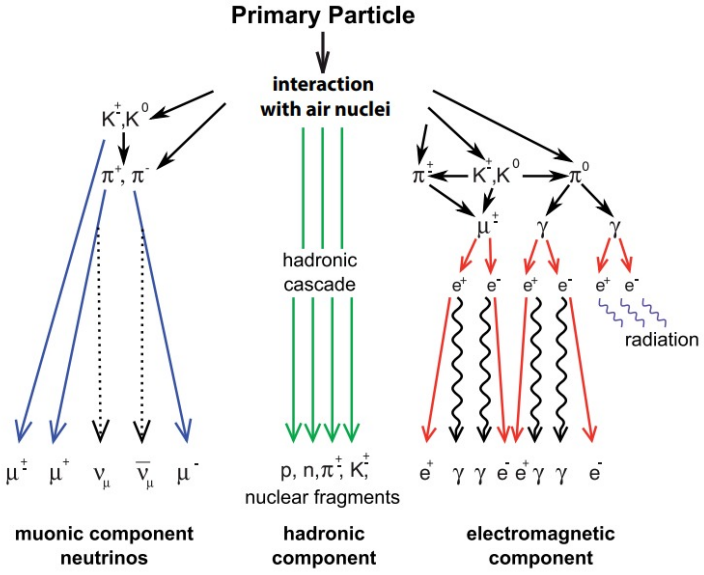

Figure 1: Illustration of an extensive air shower [4]

resulting in a cascade of particles called extensive air showers as illustrated in figure 1. An significant fraction of the particles created in primary interactions are Kaons and pions who in turn have a high likelihood of decaying into muons via the channels [2]:

$$\pi^{\pm} \to \mu^{\pm} + \nu_{\mu}(\bar{\nu}_{\mu})(BR \approx 100\%) \tag{1}$$

$$K^{\pm} \to \mu^{\pm} + \nu_{\mu}(\bar{\nu}_{\mu})(BR \approx 68\%) \tag{2}$$

In IceCube, neutrinos are detected indirectly through the detection of the leptons they produce upon interaction with the ice. However, the presence of atmospheric muons

poses a challenge, as their event rate far surpasses that of neutrino events by several orders of magnitude, creating a significant background noise.

Because neutrinos interact with matter very rarely the earth can be used as a shield ffor neutrinos as muons will either interact with earth or decay before reaching the detector. Another possible measure to distinct neutrino events from atmospheric muons is to search for events that started within the detector volume.

2.2 IceCube Neutrino Observatory

An example of scientific success in this field is the IceCube Neutrino Observatory at the South Pole. It spans an area of $7\,\mathrm{km}^3$ and is constructed solely for the purpose of detecting neutrinos from unknown sources and studying cosmic rays. The observatory observes both neutrinos and atmospheric muons, which differ from each other in their energy distribution.

Conventional muons and neutrinos are produced from the decay of longer-lived mesons like pions and kaons, which lose energy before decaying due to their longer lifetimes. This results in a steeper energy spectrum ($\propto E^{-3.7}$) compared to the spectrum of charged cosmic rays ($\propto E^{-2.7}$).

In contrast, high-energy interactions also generate short-lived heavy hadrons such as D mesons and Λ_c baryons. These decay rapidly without significant energy loss, resulting in prompt muons and neutrinos that mimic the spectrum of the charged cosmic rays they originate from.

Cosmic neutrinos and cosmic rays, in essence, have no electric charge and very small mass. Neutrinos at IceCube are detected through the interactions produced by them in the ice, which can be either charged or neutral current interactions. This leads to the release of secondary particles which radiate Cherenkov light. The light is detected by deep-embedded arrays of digital optical modules.

2.3 Design of IceCube

The IceCube Detector design detects this faint Cherenkov light coming from both neutrinos as well as atmospheric muons interacting in the detector. These sensors are mounted on 86 strings within a cubic kilometer of ice, starting down to 1450 meters in depth. Furthermore, 5160 photomultipliers are installed in this array, with a special subarray named DeepCore designed for low-energy neutrinos. It uses a denser configuration of sensors to provide higher sensitivity. Hence, the overall IceCube observatory is designed to indirectly detect and study neutrinos through their interactions within a cubic kilometer of clear Antarctic ice. The Detector is illustrated in pictur 2

2.4 Location of IceCube Observatory

The location of the IceCube Observatory at the South Pole provides several advantages:

- **Minimizes Background Noise:** The Earth's mass acts as a natural shield, reducing the number of atmospheric muons that can reach the detector from below,

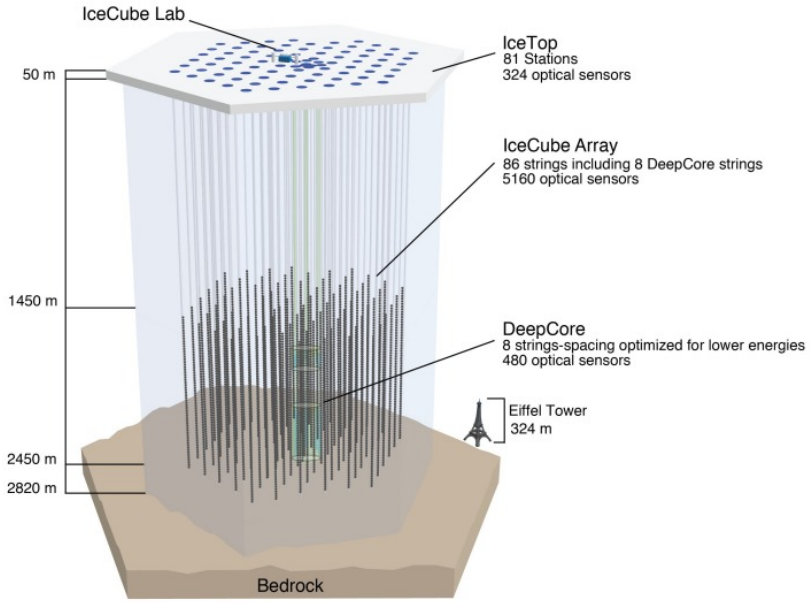

Figure 2: Schematic pdepriction of the IceCube neutrino observatory [5]

enhancing the detection of neutrinos.

- **Stable Environment:** The South Pole provides the clearest ice for detecting faint Cherenkov light signals from neutrino interactions.

2.5 Neutrino Identification

The different Cherenkov light patterns produced by different particles are used for identification. Muon neutrinos typically create long linear tracks in the ice as the resulting muon travels a considerable distance. In contrast, electron neutrinos produce more spherical light patterns known as cascades due to the localized energy deposition of the resulting electron. Tau neutrinos are challenging to detect because they decay rapidly into other particles but can produce distinctive signatures under certain conditions.

The reconstruction of these light patterns helps to determine the direction and energy of the incoming neutrinos, identifying the astrophysical sources of the neutrinos, as well as distinguishing between neutrinos of atmospheric origin and those from cosmic sources. Techniques such as Decision Trees and Naïve Bayes can be applied to improve signal-background discrimination, which is crucial for isolating astrophysical neutrino signals from other atmospheric background.

2.6 Data Preprocessing in IceCube Neutrino Detection

In our analysis of Monte Carlo simulated data from the IceCube experiment, several essential preprocessing steps were carried out to prepare the data for subsequent machine

learning tasks.

1. **Data Loading and Initial Examination:** The signal and background datasets were loaded separately to maintain the distinction between the signal (astrophysical neutrinos) and background events (muons). Basic information about the datasets, including the number of entries and attributes, was obtained using the `info` and `head` methods. This initial method helped in understanding the structure and content of the data to see if both datasets can be merged without any problems.

2. **Merging and Shuffling Data:** After confirming the structure of the data, we merged the datasets into a single dataframe. The merged dataset was then shuffled to ensure that the order of entries did not introduce any bias while applying ML algorithms.

3. **Handling Missing and Non-Numeric Values:** Entries containing only NaNs or missing values were checked and handled appropriately.

4. **Dropping Monte Carlo Truths:** Features containing Monte Carlo truths were removed to ensure that machine learning algorithms do not use information available only in simulations but irrelevant in real-world scenarios. For example, event IDs used in simulations do not provide meaningful insights for actual data analysis.

2.7 Machine Learning Methods

2.7.1 Feature Selection

After the data is preprocessed, the number of features used in the analysis has to be reduced. This can avoid unneccessary computational effort for features that do not contribute much to the classification. Also many common classification algorithms suffer from the so called 'curse of dimensionality' meaning they do not work well for extremely high-dimensional data as the size of the parameter space increases exponentially with dimensions.

Several methods exist, that aim to extract the best possible set of features. Here the mRMR algorithm (minimum redundancy, maximum relevance) was used. As the name implies this algorithm is design to extract features that correlate as strongly as possible with the class of an event while also having a low correlation with the other selected features to avoid unneccessary information. The mRMR selection is an iterative process based on the probability-density of the features and therefore has the upside of beeing model independent.

2.7.2 Quality Measures

An essential metric to evaluate the performance of a classifier is the so called confusion matrix. This is a categorization of the events classification versus their true label as true positive (tp), true negative (tn), false positive (fp) or false negative (fn) where signal

	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

Figure 3: Confusion matrix [6]

is considered a positive as depicted in figure 3. From those values two major quality measures can be constructed. The recall:

$$p = \frac{tp}{tp + fn} \tag{3}$$

indicating the fraction of actual signal events classified as signal and the precision:

$$p = \frac{tp}{tp + fp} \tag{4}$$

indicating the fraction of all events classified as signal that are actually signal. The f_β score is a a measure that provides a weighted combination of both precision and recall:

$$f_\beta = (1 + \beta^2)\frac{pr}{\beta^2 p + r} \tag{5}$$

In signal-background separation the focus is mostly on creating a very pure sample so a high precision is more important in this context.

A different way to assess the quality of a classifier is the roc-curve. This curve is created by plotting the true positive rate (TPR) versus the false positive rate (FPR) at classifiers threshold values:

$$FPR = \frac{fp}{fp + tn} \tag{6}$$

of a classifier depending on the classifiers threshold value. The area under the resulting curve equals one for a perfect classifier.

2.7.3 Gaussian Naive Bayes

The Bayes classifier is based on Bayes law on conditional probabilities:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{7}$$

Following this equation one can compute the probability $P(A|B)$ that an event belong to class A given a set of measurements B. The necessary probabilities on the right side of the equation can be obtained by estimating the probability density functions from the training data. The most common example of a bayes classifier assumes an underlying gaussian distribution. To reduce computational effort it can be useful to also assume that the features given to the classifier are statistically independent. This is a very strong assumption and therefore not always practical.

2.7.4 Random Forest

A binary Decision Tree is a classifier that works by iteratively splitting the data based on the individual features to create subsets or 'leafs' of the tree until a certain size of the tree is reached or all leaves only contain one class. The best possible split according to the training data is chosen by optimizing a loss function like the gini index. The better a split on an attribute is the more dominant is one of the classes in the resulting subset.

By training a multitude of decision trees on different subsets of the training data and averaging the results the model accuracy can be increased while also reducing the risk of overtraining. The resulting ensemble classifier is called a Random Forest.

2.7.5 KNN

Another aproach to classification is the k-nearest-neighbors classifier. In this algorithm the class of an event is decided based on the classes of the k training events that are closest to it in the data space. Because the knn-classifier relies on some kind of distance measure it may be dominated by large features unless the data is scaled before training.

2.7.6 Crossvalidation

A useful technique when assessing the quality of a machine learning model is the cross-validation. In this process the data is split into k subsets and one of the subsets is used for testing, the other ones for training. This is useful to get very precise testing results even for small datasets but can lead to high computational effort because the model has to be trained multiple times.

3 Evaluation

3.1 Feature election using the mRMR Algorithm

The following 10 features are selected by the mRMR algorithm:

1. LineFit_TTParams.lf_vel_z
2. SplineMPEDirectHitsC.n_dir_strings
3. HitStatisticsValues.z_travel
4. SplineMPEFitParams.rlogl
5. SplineMPECharacteristics.avg_dom_dist_q_tot_dom
6. LineFit_TT.zenith
7. SplineMPEDirectHitsA.n_dir_strings
8. MuEXAngular4.zenith

9. NewAtt.SplineVerRadius

10. SplineMPEMuEXDifferential.zenith

The feature selection returned the same 10 features on six different subsets of the training data amounting to a Jaccard-Index of 1.0 indicating a high stability of the selection. Despite mRMRs target to avoid redundand features there are three different estimations of the zenith angle. This speaks to the high importance of the zenith angle with regards to the classification task that is to be expected from the theory.

3.2 Classification using a Gaussian Naive Bayes

Given the 10 features chosen previously the Gaussian Naive Bayes produces the following values for the quality measures:

$$\begin{aligned} p &= 0,87 \pm 0,04 \\ r &= 0,769 \pm 0,005 \\ f_\beta &= 0,87 \pm 0,04 \\ roc\_auc\_score &= 0,92 \pm 0,06 \end{aligned}$$

Those values are obtained using 5-fold cross-validation. A roc curve for a possible train-

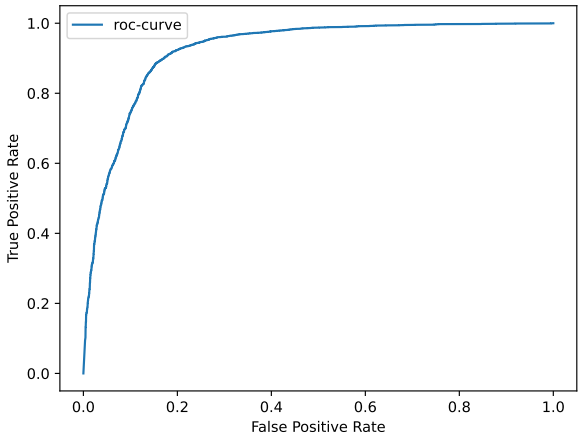

Figure 4: Roc-curve for a gaussian naive bayes classifier based on a train-test-split with 80% of data used for training

test split with a ratio of 20% test data is depicted in figure 4. Considering the strong assumption of independence between the features the different estimations of the zenith angle can pose a challenge in this context. After eliminating two of the three estimations

linked to the zenith angle the quality of the reconstruction improves:

$$\begin{aligned} p &= 0,893 \pm 0,035 \\ r &= 0,788 \pm 0,005 \\ f_\beta &= 0,892 \pm 0,034 \\ roc\_auc\_score &= 0,93 \pm 0,04 \end{aligned}$$

by approximately 2% with regards to precision and recall despite the reduced amount of features. The improved roc-curve can be seen in Figure 5.

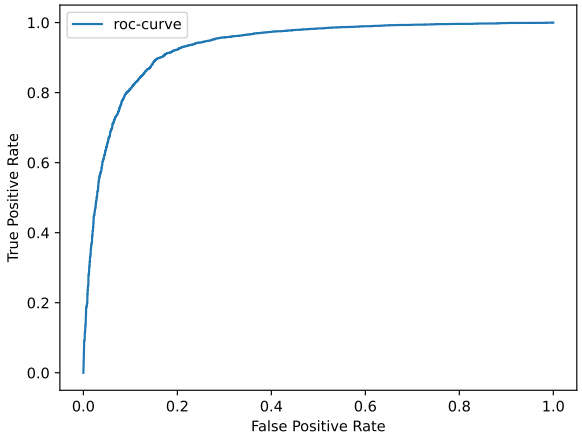

Figure 5: Roc-curve based on a train-test-split with 80% of data used for training without the redundand zenith-angle-related features

3.3 Classification using a Random Forest

Like the gaussian Naive Bayes the Random orest is trained and then evaluated based on a 5-fold-cross-validation. The resulting quality parameters are:

$$\begin{aligned} p &= 0,945 \pm 0,023 \\ r &= 0,9242 \pm 0,0032 \\ f_\beta &= 0,945 \pm 0,022 \\ roc\_auc\_score &= 0,981 \pm 0,009 \end{aligned}$$

These show a significant improvement in precision and even more in recall compared to the Naive Bayes model. This is also clearly visible when it comes to the roc-curve **??**roc_rf. The reduced errors on the quality measures show the increased stability of the model thanks to the bagging of multiple classifiers.

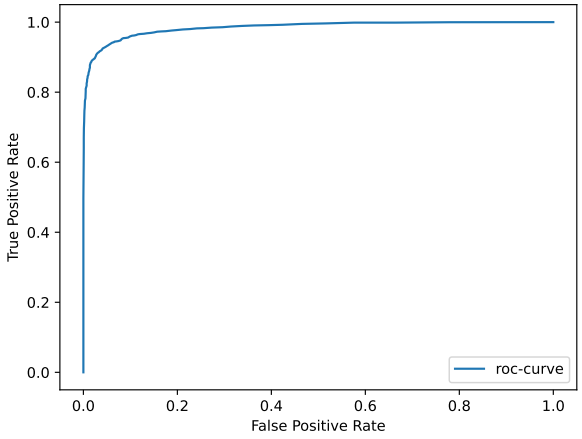

Figure 6: Roc-curve based on a train-test-split with 80% of data used for training for a random forest classifier and 10 features

3.4 Classification using a K-Nearest-Neighbors Model

At last a KNN model with the parameter $k = 8$ was trained for the same signal-background-selection. As mentioned in the theory section the knn classifier unlike the

Table 1: Crossvalidated quality measures for the KNN classifier depending on the scaling

scaling	p	r	f_{beta}	roc_auc_score
none	0,9±0,04	0,858±0.005	0,9±0,04	0,93±0,04
standard	0,937±0,024	0,904±0,004	0,937±0,024	0,968±0,016

previous two models is very susceptible to domination by large attributes. Therefore as table 1 showcases the model performs significantly better when the underlying attributes are first rescaled to center around zero with a standard deviation of $\sigma = 1$. The corresponding roc-curves are shown in figure 7. A difference in the prediction of the knn classifier can be seen when looking at the distributions of the probabilities predicted by the individual classifiers. As figure 8 shows the knn algorithm has a lot more events predicted around a value of $0,5$ than the other classifiers. This can be used by cutting out values who are predicted with a low accuracy to either side. Applying this to a standard 80% train-test-split results in an improved precision of $p = 0.97$.

3.5 Variation of the threshhold

Another measure to improve the performance of a classifier is to change the threshhold value. To investigate this the f_β-score with $\beta = 0.5$ is calculated depending on the threshhold for the random forest classifier. The result is displayed in Figure 9. The

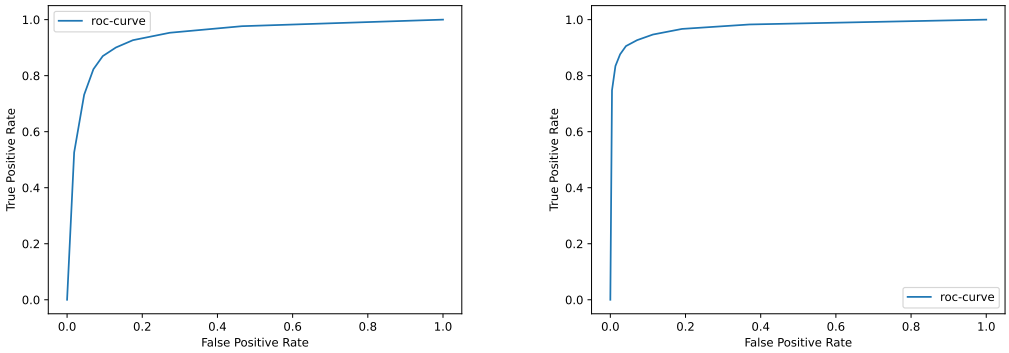

Figure 7: Roc-curve based on a train-test-split with 80% of data used for training for a knn classifier and unscaled data (left) and scaled data (right)

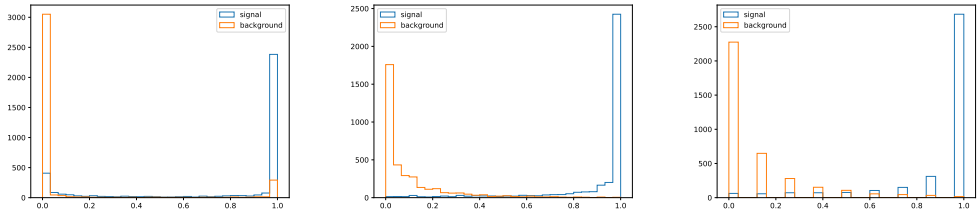

Figure 8: Distribution of the probabilities predicted by the naive bayes (left), random forest (middle) and knn (right)

maximum value of $f_{0.5} = 0.954$ is reached for a threshhold value of $T = 0.65$. This Threshhold reaches an improved precision of $p = 0.978$ and a reduced recall of $r = 0.89$.

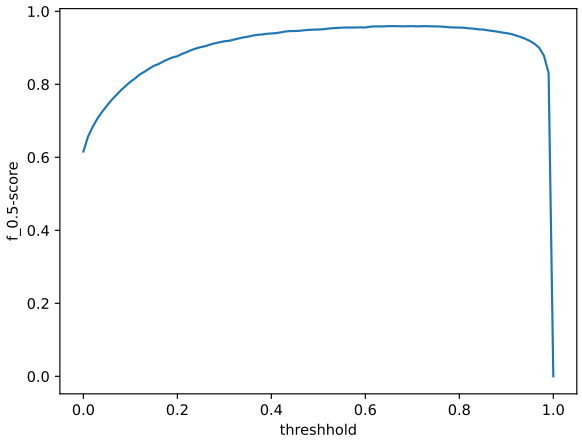

Figure 9: F-score depending on the classifier threshhold for the random forest classifier

4 Discussion

While all three methods of classification perform resonably well on the features selected by the mRMR algorithm there are distinct differences in quality. The Gaussian Naive Bayes classifier performs the worst in terms of all relevant quality measures. Especially the recall is very low at $76,9\%$ compared to $90,4\%$ for the knn classifier and $92,2\%$ for the random forest. This might be due to low complexity of the model because of the underlying assumption of statistical independence. Here further changes on the selected features to reduce correlation amongs the features could improve performance.

The random forest performs very well in terms of precision $(94,6\%)$ and roc-auc-score $(0,981)$ outperforming both other classifiers in all relevant quality metrics. It also has the lowest errors in the quality measures proving the models high stability when tested in crossvalidation. This comes at the expense of higher computational effort because the bagging method requires a high number of different classifiers to be trained. While this might pose a problem in different applications in this case the amount of data is too low for this to have a significant impact.

The knn classifier significantly outperforms the naive bayes model with a high precision of $93,7\%$ when using scaled data. While it does not quite reach the performance of a random forest it has the advantage of beeing easily adapted upon the introduction of new training data. It also more easily enables the elimination of badly predicted events, which comes at the cost of reducing the amount of available data but can improve precision.

References

[1] *Data Analysis with IceCube Monte Carlo Simulation Data*. Fakultät Physik, TU Dortmund. 2024.

[2] Thomas K. Gaisser, Ralph Engel, and Elisa Resconi. *Cosmic Rays and Particle Physics*. 2nd ed. Cambridge University Press, 2016. DOI: 10.1017/CBO9781139192194.

[3] Andreas Haungs, Heinigerd Rebel, and Markus Roth. "Energy spectrum and mass composition of high-energy cosmic rays". In: *Reports on Progress in Physics* 66.7 (July 2003), pp. 1145–1206. DOI: 10.1088/0034-4885/66/7/202.

[4] Andreas Haungs et al. *The KASCADE Cosmic-ray Data Centre KCDC: Granting Open Access to Astroparticle Physics Research Data*. June 2018.

[5] Hermann Kolanoski and Norbert Wermes. "1Introduction". In: *Particle Detectors: Fundamentals and Applications*. Oxford University Press, June 2020. ISBN: 9780198858362. DOI: 10.1093/oso/9780198858362.003.0001. eprint: https://academic.oup.com/book/0/chapter/365027446/chapter-pdf/50270332/oso-9780198858362-chapter-1.pdf. URL: https://doi.org/10.1093/oso/9780198858362.003.0001.

[6] Prof. Dr. Emmanuel Müller. "Big Data Analytics, Chapter 5: Classification". In: ().