**Exploring the intersections between Information Visualization
and Machine Learning**

Igor Bueno Corrêa

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências
de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Igor Bueno Corrêa

Exploring the intersections between Information Visualization and Machine Learning

Master dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC- USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho

**USP – São Carlos
December 2018**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

B928e Bueno Corrêa, Igor
 Exploring the intersections between Information
Visualization and Machine Learning / Igor Bueno
Corrêa; orientador André Carlos Ponce de Leon
Ferreira de Carvalho. -- São Carlos, 2018.
 79 p.

 Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2018.

 1. Information Visualization. 2. Machine
Learning. 3. RadViz. 4. Visual Analytics. I. Carlos
Ponce de Leon Ferreira de Carvalho, André, orient.
II. Título.

Igor Bueno Corrêa

**Explorando as interseções entre Visualização da
Informação e Aprendizado de Máquina**

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho

**USP – São Carlos
Dezembro de 2018**

ACKNOWLEDGEMENTS

Agradeço primeiramente ao meu orientador Prof. André pelas palavras de apoio e por acreditar em mim;

à minha mãe e minha irmã por todo o amor que recebo e pelo apoio incondicional nos momentos difíceis desta jornada acadêmica;

à Jessica e Yuri pela companhia, hospitalidade e incentivo;

aos amigos do Microsegundo sem os quais tudo seria mais difícil;

ao time do Ateliê de Software de Poços de Caldas pela compreensão e apoio;

à psicóloga Fernanda Torraca por me ajudar a encontrar a força que precisava;

à banca examinadora pelas valiosas sugestões;

ao CNPq pelo apoio financeiro.

ABSTRACT

CORRÊA, I. B.. **Exploring the intersections between Information Visualization and Machine Learning**. 2018. 79 f. Master dissertation (Master student em Program in Computer Science and Computational Mathematics) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

With today's flood of data coming from many types of sources, Machine Learning becomes increasingly important. Though, many times the use of Machine Learning is not enough to make sense of all this data. This makes visualization a very useful tool for Machine Learning practitioners and data analysts alike. Interactive visualization techniques can be very helpful by giving insight on the meaning of the output from classification tasks. In this work, the aim is to explore, implement and evaluate different visualization techniques with the explicit goal of directly relating these visualization to the Machine Learning process. The proposed approach is the development of visualization techniques for *a posteriori* analysis that combines data exploration and classification evaluation. Results include a modified version of the Radial Visualization technique, called Dual RadViz, and also the use of interactive multiclass Partial Dependence Plots as means of finding counterfactual explanations about Machine Learning classification. An account of some of the many ways Machine Learning and visualization are used together is also given.

Key-words: machine learning, information visualization, RadViz, visual analytics.

RESUMO

CORRÊA, I. B.. **Exploring the intersections between Information Visualization and Machine Learning**. 2018. 79 f. Master dissertation (Master student em Program in Computer Science and Computational Mathematics) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Hoje em dia, com o enorme fluxo de dados provenientes de muitos tipos de fontes, Aprendizado de Máquina se torna cada vez mais importante. No entanto, muitas vezes o uso de Aprendizado de Máquina não é o suficiente para que seja possível enxergar o valor e o significado de todos estes dados. Isso faz com que visualização seja uma valiosa ferramenta tanto para analistas de dados quanto para aqueles que praticam tarefas relacionadas à Aprendizado de Máquina. Técnicas de visualização interativa podem ser de grande utilidade por possibilitarem *insights* sobre o significado do resultado de tarefas de classificação. Neste trabalho, o objetivo é explorar, implementar e avaliar diferentes técnicas de visualização, explicitamente focando em suas relações com o processo de Aprendizado de Máquina. A abordagem proposta se trata do desenvolvimento de técnicas de visualização para análise *a posteriori* dos resultados de tarefas de classificação, combinando avaliação da classificação e exploração visual de dados. Os resultados incluem uma versão modificada da técnica de Visualização Radial, chamada Dual RadViz, e também o uso de Gráficos de Dependência Parcial multiclasse interativos como meio de se chegar à explicações contrafatuais sobre resultados de classificação. É dado também um relato de algumas das muitas maneiras onde Aprendizado de Máquina e visualização são usados conjuntamente.

Palavras-chave: aprendizado de máquina, visualização da informação, RadViz, visual analytics.

LIST OF FIGURES

Figure 1 – The Visual Analytics process	22
Figure 2 – Scatterplot matrix	39
Figure 3 – t-SNE applied to the MNIST dataset	39
Figure 4 – Basic RadViz illustration	41
Figure 5 – Example of Parallel Coordinates Visualization	42
Figure 6 – Infuse - scatterplot view	44
Figure 7 – EnsembleMatrix interface	45
Figure 8 – FCWL interface	46
Figure 9 – Confusion matrix comparison	48
Figure 10 – Neighbor Joining Trees	48
Figure 11 – Confusion wheel	49
Figure 12 – IFM - example of SOM-based visualization	50
Figure 13 – Intelligent, high-dimensional volume classification	51
Figure 14 – Visualization of Deep Convolutional Neural Networks	52
Figure 15 – Synthetic images for CNN Visualization	53
Figure 16 – Decision boundaries for SVM classifier	53
Figure 17 – Partial Dependence Plot	54
Figure 18 – Analysing partial dependence with Prospector	55
Figure 19 – Slider position and influence of each anchor group	59
Figure 20 – Dual RadViz with examples from the Iris Dataset	59
Figure 21 – Illustrative example of <i>confusion colors</i>	60
Figure 22 – Multiclass PDP examples	62
Figure 23 – WBVA overview	63
Figure 24 – Illustration of Dual RadViz settings	64
Figure 25 – New Thyroid dataset classified with Model Averaged Neural Network	69
Figure 26 – PDP on dimension T3resin for misclassified example	70
Figure 27 – Different visualizations through the use of the slider for the Vertebral Column dataset	71
Figure 28 – PDP on dimension <i>pelvic incidence</i>	71
Figure 29 – PDP on dimension <i>sacral slope</i>	72

LIST OF TABLES

Table 1 – Confusion matrix	33
Table 2 – Reviewed works	56

LIST OF ABBREVIATIONS AND ACRONYMS

CG		Class Group
CNN		Convolutional Neural Network
CSV		Comma-separated values
D3		Data-Driven Documents
DG		Dimension Group
DH		Disk Hernia
DM		Data Mining
FCWL	...	First Certain Wrong Labeled
FN		False Negative
FP		False Positive
GDPR		General Data Protection Regulation
ICA		Independent Component Analysis
IFM		Image Feature Map
KDD		Knowledge Discovery in Databases
ML		Machine Learning
NO		Normal
PCA		Principal Component Analysis
PDP		Partial Dependence Plot
RadViz	...	Radial Visualization
SAR		Synthetic Aperture Radar
SL		Spondylolisthesis
SOM		Self-Organizing Map
SVM		Support Vector Machines
t-SNE		t-Distributed Stochastic Neighbor Embedding
TN		True Negative
TP		True Positive
TSP		Traveling Salesman Problem
VDM		Visual Data Mining

CONTENTS

1	INTRODUCTION	21
1.1	The need for transparency on Machine Learning solutions	22
1.2	Hypothesis	23
1.3	Aims and objectives	23
1.4	Organization of the work	24
2	MACHINE LEARNING OVERVIEW	27
2.1	Main types of Machine Learning	27
2.1.1	<i>Unsupervised learning</i>	27
2.1.2	<i>Supervised learning</i>	28
2.1.3	<i>Semi-supervised learning</i>	28
2.2	Predictive Machine Learning tasks	28
2.2.1	<i>Predicting classes</i>	28
2.2.2	<i>Predicting real values</i>	29
2.2.3	<i>Predicting probabilities</i>	29
2.3	The main Machine Learning steps	29
2.3.1	<i>Before training: Data preprocessing</i>	29
2.3.2	<i>After training: Experiments and performance evaluation</i>	31
2.3.2.1	<i>Experiments</i>	32
2.3.2.2	<i>Performance evaluation</i>	32
2.4	Final considerations	33
3	VISUALIZATION OVERVIEW	35
3.1	Perception and visual variables	36
3.2	Techniques for multivariate data	38
3.2.1	<i>Scatterplot matrix</i>	38
3.2.2	<i>t-SNE</i>	39
3.2.3	<i>RadViz</i>	40
3.2.4	<i>Parallel Coordinates Visualization</i>	41
3.2.5	<i>Other techniques</i>	41
3.3	Final considerations	42
4	THE INTERSECTION OF MACHINE LEARNING AND VISUAL- IZATION	43

4.1	Visualization in data preprocessing	44
4.2	Visualization assisting model creation	45
4.2.1	<i>Visualization in active learning</i>	45
4.3	Visualization to explore classification results	47
4.4	ML to enable or improve visualization	49
4.5	Visualization to provide insight into black-box ML algorithms	51
4.5.1	<i>Prospector and the used of partial dependence plots</i>	53
4.5.1.1	<i>Partial Dependence Plots</i>	54
4.5.1.2	Prospector	55
4.6	Summary of reviewed works	55
4.7	Final considerations	55
5	METHODS: WBVA	57
5.1	Dual RadViz: A modification to the RadViz technique	58
5.2	The <i>Confusion Colors</i> color scheme	60
5.3	Combination of Parallel Coordinates Visualization and RadViz	61
5.4	Multiclass Partial Dependence Plots as means of interaction	61
5.5	WBVA environment	62
6	RESULTS	65
6.1	Datasets	65
6.2	Algorithms used for training	66
6.3	Usage scenarios	66
7	CONCLUSION, LIMITATIONS, AND FUTURE WORK	73
	BIBLIOGRAPHY	75

Explorando as interseções entre Visualização da Informação e Aprendizado de Máquina

INTRODUCTION

In the world we live in a vast amount of data is produced every day by virtually every sector of our society. Data can come from industry, business, measurements from the environment, scientific experiments and directly from people (via social networks such as Facebook or Twitter, for instance). With current technology we are able to store this data, but it is usually unstructured, unfiltered and not ready to be used in valuable ways. It is clear that valuable information is contained in all this data, but the data itself does not have any value without analytical methods that enable decision-makers to use it. Without proper methods to extract value from data, opportunities, money and time can be wasted.

As our ability to produce and store data increases faster than our ability to process and extract valuable information from it, we are faced with the information overload problem ([KEIM *et al.*, 2010](#)), which refers to:

- Getting lost in data that is irrelevant to accomplish whatever task is at hand;
- Data being processed in inappropriate ways;
- Information being presented in ineffective ways.

The field of Machine Learning addresses some aspects of this problem by providing automated ways to acquire useful information from data. Through Machine Learning techniques, patterns can be extracted and predictive models can be created in order to help people and organizations make better decisions.

Despite its success, Machine Learning has some limitations regarding the communication of the knowledge generated from data to the user. If decisions are made based on automated processes and these decisions turn out to be wrong, it becomes very important to further examine these processes. Also, many tasks cannot be fully automated, and the user should be able to input as much of his or her knowledge into the process as possible.

For the user to better understand the data at hand and better interact with the results from models generated from data, Information Visualization is a valuable tool. By visualizing the data itself and also the results from Machine Learning techniques, the abilities of the human visual and cognitive systems can be harnessed and become valuable in the process of knowledge discovery. For instance, it has been shown that preattentive processing capabilities of the visual system allows humans to detect features in images in extremely short time (HEALEY *et al.*, 1995).

In this context, the field of Visual Analytics provides methods for integrating visual and automated processes for data exploration and analysis.

Figure 1 – The Visual Analytics process

Source: Keim *et al.* (2010).

In Visual Analytics, knowledge acquisition from data is an iterative process. The loop involving visual data exploration and automated data analysis is shown on Figure 1. The arrows between Visualization and Models indicate some of the main ways Machine Learning and Visualization interact: the user can visualize the current model and visualization can guide the process of building it.

1.1 The need for transparency on Machine Learning solutions

Transparency on Machine Learning solutions is a current relevant issue, as laws start to try and implement policies around automated decisions that directly affect human beings. As an example, the General Data Protection Regulation (GDPR), which is a set of regulation laws

recently put into place in the European Union, mentions that subjects of automated decisions have a "right to explanation" (SELBST; POWLES, 2017). Typically, transparency in the context of human interpretability of Machine Learning models is an explanation of "what", "how", or "why" the model arrived at a specific outcome. Weller (2017) enumerates several objectives of transparency, such as to provide a user with an explanation why a certain prediction or decision was made, and to allow the system to be checked as to whether or not it worked appropriately. This sort of transparency could be the basis for the individuals to challenge automated decisions affecting them. The importance of this kind of transparency is evident if we consider cases where the prediction is consequential to the human subject, such as criminal sentencing, job application or credit approval.

One such kind of transparency could take the form of *counterfactual explanations* (WACHTER *et al.*, 2017). An example of a counterfactual explanation could be: "You were denied a loan because your annual income was £30,000. If your income had been £45,000, you would have been offered a loan." In such explanation the decision is stated and then it is contrasted with a counterfactual, that is, how the world would have to be different in order for another outcome to occur, presumably a more desirable one. This can be a valuable explanation and is independent of the complexity of the system that made the decision.

Systems that use visualization have tackled the challenge of transparency, as will be discussed on [section 4.5](#).

1.2 Hypothesis

Hypothesis 1 Animated transitions between visualization of classification results and visualization of data attributes can aid the process of understanding the created model.

Hypothesis 2 Interactive Partial Dependence Plots can be used to provide useful explanations for ML classification results.

1.3 Aims and objectives

The general aim of this work is to explore the many ways Machine Learning and Visualization can be used together in order to generate useful knowledge from data. Specific objectives are:

1. To survey the different objectives and cases where Information Visualization and Machine Learning have been used together;
2. Provide to both domain experts and non-experts a system for visual and interactive data exploration and classification results analysis, complemented by traditional metrics;

3. Through this system, provide a means of analysis of the impact that individual attributes have on the behavior of a given classification model.

Relating to objective 2, tasks the user should be able to perform include:

- Get a sense of class separation within the dataset;
- Get a sense of how particular attributes correlate to the examples' true classes;
- Identify examples misclassified with high confidence or correctly classified with low confidence.

Relating to objective 3, some tasks are:

- Inspect if by changing the value of an attribute for an example or set of examples, the result of the classification output changes;
- Identify attributes that do not impact significantly the classification result, regardless of their values.

Objectives 2 and 3 also refer to improving the user's confidence in the model. This is related to the interpretability and transparency of the model's result. By knowing why a given result was returned by a Machine Learning model, the user has a better chance of knowing whether or not the model's output makes sense ([KRAUSE *et al.*, 2016](#)).

1.4 Organization of the work

In order to contextualize the exploration of instances where Machine Learning and Information Visualization come together, this work has the following structure:

In [chapter 2](#) an overview of Machine Learning is given. This chapter identifies some of the types and tasks of Machine Learning. Concepts such as data preprocessing and model validation are also discussed.

In [chapter 3](#) an overview of the visualization field is given. Which visual variables are available to be used in visualization as well as some specific techniques are presented. Particular emphasis is given to the RadViz technique, since it plays a major role on the proposed system.

After this contextualization, [chapter 4](#) discusses specific ways in which Machine Learning and Visualization interact. This chapter presents many examples of recent works in which this interaction is present. The concept of partial dependence and the related Partial Dependence Plot is given particular attention, since it is used in the proposed system, relating specifically to objective 3.

In [chapter 5](#) the methods and techniques implemented to achieve objectives 2 and 3 discussed in detail, and on [chapter 6](#) the results are shown with usage examples.

Conclusion, limitations and possible future work are also addressed in [chapter 6](#).

MACHINE LEARNING OVERVIEW

The task of acquiring knowledge from data involves many approaches and techniques. For instance, Data Mining (DM) and Knowledge Discovery in Databases (KDD) regard this process as an exploratory one, where previously unknown characteristics and patterns hidden in the data can be discovered. In Machine Learning (ML) the goal is usually to acquire knowledge from a sample of the data in a way that this knowledge can be applied to unseen data. In other words, ML is specially concerned in learning patterns from a given dataset so that this knowledge can be generalized (ZHOU; CHEN, 2015). Even though this definition may seem simple, there are many steps that have to be carried out in order to successfully accomplish this goal. In this chapter some of the challenges, applications and methods of ML are discussed.

2.1 Main types of Machine Learning

In ML, the main approaches for learning from data are: *unsupervised learning*, *supervised learning*, *semi-supervised learning* and *reinforcement learning*. Here the first three of the mentioned approaches are briefly explained (ABU-MOSTAFA *et al.*, 2012).

2.1.1 *Unsupervised learning*

In this type of learning the data used in the process is not labeled (has no predefined class) and the goal is to learn its structure. Clustering algorithms can do this by separating the data into groups that are internally similar and different from each other. Depending on the nature of the data, it could make sense to divide these groups further into sub-groups, in which case the process is called hierarchical learning.

In business, an example of application would be to divide customers into groups so that advertising could be more effective, as it would be targeted to customers with similar profiles.

2.1.2 Supervised learning

In supervised learning the data used has labels. The goal is to induce a model that learns from a set of examples, each comprised of attribute values plus labels. This model should then be able to classify new unlabeled examples as correctly as possible. Classification is a very common task and an example is the detection of spam e-mail: given the content of a new e-mail message, should it be classified as spam or not?

In [section 2.2](#) different supervised learning tasks are briefly discussed, and methods to perform these tasks are mentioned.

2.1.3 Semi-supervised learning

In order to perform supervised learning, every example used in the learning process must be labeled. This can be a problem in many areas, as it might be very costly to label the data. The sheer amount of data is often overwhelming, making the task of manually labeling every instance completely unfeasible. Semi-supervised learning allows the generation of a predictive model that, by the added use of unlabeled examples in the learning process, is better than models that could be induced by only using labeled data. Most semi-supervised learning methods consist in iterative automated labeling of unlabeled examples, so that these newly labeled examples can be used in the training process ([ZHU, 2011](#)).

2.2 Predictive Machine Learning tasks

It was mentioned that supervised learning can be used to induce a classifier. But there are other kinds of outputs that might be desirable from the induced model besides a class. Different algorithms can be used to model a real valued function or a probability distribution.

2.2.1 Predicting classes

In this task, a model is trained based on some data, such that given some new example, it outputs which class this example probably belongs to. Considering $\mathcal{X}$ as the set of all possible examples and $\mathcal{Y}$ the set of all possible classes, it is assumed that there is a function $f : \mathcal{X} \rightarrow \mathcal{Y}$, that is, a function that maps any given $\mathbf{x}$ from $\mathcal{X}$ to a corresponding $\mathbf{y}$, from $\mathcal{Y}$.

If $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ is the training dataset, the goal is to produce a function $g(\mathbf{x}) = \mathbf{y}$ for an $\mathbf{x}$ not contained in $\mathcal{D}$. Algorithms such as Support Vector Machines (SVM), Neural Networks and Decision Trees can be used to perform such tasks.

2.2.2 Predicting real values

For some problems the output of the induced model should be a real number. An example would be credit application. When a new client applies for credit at a bank, a model could be induced from data from all current and past clients. This model could output if the client is suitable for getting the credit or not. This would be a classification problem, where the classes could be “*approved*” or “*denied*”. Alternatively, the model could output how much credit would be appropriate to concede to this new client. So the model must learn a function that, given the data regarding a new client, returns a real number representing the amount of money this client can apply for.

For such task Linear Regression can be employed. Though it’s very easy to train, the main limitation of this method is that, if the function to be learned is not linear, the results might be exceedingly inaccurate. Other algorithms, such as Neural Networks, can be adapted to this task as well.

2.2.3 Predicting probabilities

Logistic Regression is an example of algorithm that can be used to learn a probability distribution $P(\mathbf{y}|\mathbf{x})$. That is, given some input $\mathbf{x}$, what is the probability of outcome $\mathbf{y}$. One example of a problem suitable for this approach in medicine is the estimation of risk of some of disease. It is useful to estimate, for instance, what is the probability of a given patient having cancer in the next decade, given some information about him or her, such as medical history, eating habits, etc. As with regression, other algorithms can be adapted to this task as well.

2.3 The main Machine Learning steps

The focus up to this point was mainly on one step of the learning process, which is the induction of the model. Invariably though, other steps must be taken in order to achieve successful learning. In this section these steps are discussed in a standard order. It should be noted that the actual process does not need to be strictly in this order, and that it is very common to iterate through them before generating an effective model (WITTEN *et al.*, 2011).

2.3.1 Before training: Data preprocessing

Ideally, the data used in the learning process would be structured and reliable. In real scenarios though, many factors can make the data unreliable. The data itself can be noisy, sensors might be faulty and generate anomalous examples, too many values might be missing, the data might be comprised of a lot of redundant information, etc. Therefore, real world data is usually inadequate to be used in the learning process without some sort of preprocessing. In this sub-section some of the most common preprocessing tasks are discussed.

Missing values Many strategies can be used to deal with missing values. A simple one would be to simply ignore the example if it has too many missing values. If the problem seems related to one data attribute in particular, the attribute can be ignored for all examples. Alternatively, the attribute can be converted to binary where 1 means the attribute was present, and 0 means it was not.

Another approach is to estimate these missing values. For numerical attributes, the average value can be used. As a refinement, the average value only considering the examples of the same class as the example with the missing value can be used. For nominal attributes the mode can be used, or a new attribute value ("*missing*", for instance) can replace the missing values for all examples.

Outlier detection An outlier is a data example that significantly deviates from the others. Sometimes these examples are legitimate and therefore they might represent valuable information. But it is also common that such examples represent some kind of measurement error or noise.

The presence of outliers in the data can be very harmful to the model induction process, specially if the learning algorithm is overly sensitive to them. Identifying and deciding whether or not to eliminate such examples is an important step in the data cleaning process.

Attribute type conversion A very common preprocessing task is the conversion of attributes from one type to another. As many learning algorithms, such as Neural Networks and SVMs, can only deal with numerical attributes, one of the most common conversions is from nominal to numerical.

Attributes can be ordinal in nature. For instance, an attribute *size* could have "*small*", "*medium*" and "*large*" as possible values. In such cases, the values can be converted to integers such that the order relation is preserved.

If there is no order relation between the attribute values, canonical conversion can be employed. Each distinct attribute value generates a new binary attribute. For instance, if the attribute *color* of a given dataset can assume one of the given values { "*black*", "*white*", "*red*" }, after canonical conversion, the dataset would no longer have the *color* attribute. Instead, it would have 3 new binary attributes: *black*, *white* and *red*. For each example only one of these new attributes would have its value assigned to 1.

Class imbalance It is often the case that the distribution of classes in a dataset is not balanced. For instance, in a credit card fraud detection scenario, the examples are mostly of non-fraudulent activity. Therefore, when inducing a classification model, there would be a tendency in favor of the majority class. In this case, the results would be disastrous. The goal in such scenario is precisely to identify the cases where fraud might be happening, even if that generates some false alarms (legitimate activity classified as fraud).

There are many strategies to deal with this problem. Some of these approaches are the undersampling of the majority class, oversampling of the minority class or the generation of artificial examples similar to those of the minority class.

Dimensionality reduction Several disadvantages can come from having a dataset with very high dimensionality in Machine Learning. One of the problems is that with high dimensionality, the data tends to be sparse in the hyper-volume that contains it, and many Machine Learning algorithms would have trouble finding patterns in such sparse data. This phenomenon is known as *curse of dimensionality* (TAN *et al.*, 2005).

Also, a lot of redundant information might be contained in the data. Discarding this information would improve the simplicity of the generated models as well as decrease processing time and memory use.

One approach to dimensionality reduction is feature (or attribute) selection. Feature selection consists in using only a subset of the original data attributes. For instance, if some attribute is highly correlated with another, one of them can be discarded without significant information loss. The attributes can also be ranked based on how informative they are, and the most informative ones selected (WITTEN *et al.*, 2011).

Another approach is feature extraction. New attributes can be created by different combinations of the original attributes. This method can involve domain knowledge from the user or be automated. Methods such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA) map the data to a lower dimensional space while trying to preserve as much information as possible. In PCA the data is mapped to a new coordinate system in low dimension while maximizing the preservation of the variance present in the data. ICA is used to create new components that are as statistically independent from each other as possible.

A disadvantage of feature extraction is that the new set of attributes do not preserve the meaning inherent to the original ones. Sometimes it is desirable to have a clear interpretation of the results, and this might depend on the attributes' meaning.

2.3.2 After training: Experiments and performance evaluation

Given a dataset, it's not particularly difficult to model the data, as long as a complex enough learning method is employed. But the goal of supervised learning is not only to *explain* the sample data at hand. The model has to generalize to perform well in examples never seen during training. That is, the goal is to minimize the *out-of-sample* error.

If a particular model presents no error at all in the training set, this does not mean it will perform well in the test set. In fact, there is a trade off. The more the model fits the training data, the less chance it has of performing well on test data (this problem is called *overfitting*). So the problem becomes: how to estimate the out-of-sample error given that all that is available is

some sample data? To perform this validation one can run different experiments and then apply performance metrics in order to evaluate the experiments' results.

2.3.2.1 Experiments

Cross-Validation With cross-validation, the available dataset, comprised of N examples, is divided into P partitions where each partition contains N/P examples, ideally. When training the model, all but one of these partitions are used, and afterwards the remaining partition is used for testing. After doing this P times, the error rate, or other performance measurements, are averaged. In this way, all examples are used for training and all examples are used for testing.

A particular case of cross-validation is *leave one out*, where all but one example are used for training and the resulting model is tested only in the remaining one. This can be very computationally costly, and a more viable practice is to use 10-fold cross-validation, that is, cross-validation with 10 partitions.

Bootstrap Given a dataset $\mathcal{D}$ with N examples, this method creates another dataset $\mathcal{D}'$ of size N by randomly sampling $\mathcal{D}$ N times, with replacement. This contrasts with cross-validation in two ways. First, in cross-validation the training set always has less examples than the whole dataset $\mathcal{D}$. Second, in cross-validation there are no duplicates on the training set, whereas in bootstrap, as the sampling is with replacement, the training set is bound to have some duplicates. To evaluate the performance using bootstrap, the examples not picked while constructing the training set $\mathcal{D}'$ are used. For a reasonably large dataset the training set generated with bootstrap tends to have 63.2% of the examples in $\mathcal{D}$, leaving 36.8% examples for testing.

2.3.2.2 Performance evaluation

Some straightforward metrics that can be used after testing the model are *accuracy*, *precision*, *recall* and *specificity*. Accuracy is the ratio of examples correctly classified to the sum of all examples used during testing.

In order to calculate the remaining metrics the concepts of True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) need to be established. It is assumed a binary classification scenario, with the two classes being “positive” and “negative”.

TP are the examples of the “positive” class that were correctly classified as “positive”. Similarly, TN are the examples of the “negative” class that were correctly classified as “negative”. FP are the “negative” examples that were missclassified as “positive” (a false alarm), and FN are the “positive” examples missclassified as “negative”. A confusion matrix can be used to represent such concepts after classification, as shown on [Table 1](#).

Precision is the ratio of true positives to all examples classified as positive (including false alarms), that is: $precision = \frac{TP}{TP+FP}$.

Recall is the ratio of true positives to all examples that are actually positive, therefore $recall = \frac{TP}{TP+FN}$. Similarly to recall, specificity is the ratio of true negatives to all examples that are in fact negative, that is: $specificity = \frac{TN}{TN+FP}$.

Table 1 – Confusion matrix

	Predicted as positive	Predicted as negative
Actually positive	TP	FN
Actually negative	FP	TN

Source: Adapted from [Witten et al. \(2011\)](#).

One additional metric is the Kappa coefficient, which compares the observed accuracy with the expected accuracy for a random classifier.

2.4 Final considerations

This chapter broadly discussed some of the main types of ML tasks, as well as steps that should be taken before and after the learning itself. This, alongside with [chapter 3](#), gives sufficient context to [chapter 4](#), where the intersection between ML and Information Visualization is explored.

VISUALIZATION OVERVIEW

Graphical representations of many different types are used by humanity since its early history in order to communicate information. With the advent of science and the increasing availability of data coming from diverse sources, the area of Information Visualization becomes an emerging and promising field.

In order to create meaningful visualizations, it is necessary to have in mind some basic and important factors such as the cardinality of the dataset and its number of dimensions. Large volumes of data have to be somehow summarized in the visualization and too many dimensions cannot be visualized at once. It's also important to define what information has to be communicated through the visualization, and who is the target audience. Different visual metaphors might be more effective for a certain audience than others.

As defined by [Chan \(2006\)](#), Information Visualization is the use of computer-based interactive visual representations of abstract and non-physically based data to amplify human cognition. If this representation is done well, users can detect previously unknown patterns on the data, confirm some previously assumed hypothesis, or just use the visualization to present some idea to others.

Structured data is usually represented by tables and text. It is very difficult to gain insight by analyzing it through such representations. The idea of Information Visualization is to take advantage of capabilities of the human visual system. Our natural perceptual and cognitive capabilities of detecting patterns and correlations can be harnessed through a good visual representation of the data.

Some of the most basic and commonly used and techniques are:

Line graph Line graphs are primarily a 1-dimensional visualization technique. The y axis represents the range of values the examples can take and the x axis corresponds to the order of the examples themselves. It's commonly used to show changes or tendencies

through time. Line graphs can be adapted to multidimensional data by using more than one line, but the effectiveness of pattern detection can be compromised.

Bar graph Commonly used to represent quantities of different categories or groups. The dimensions of each bar, which can be represented horizontally or vertically, correspond to the values related to each category. These bars are usually 2D rectangles of equal widths.

Frequency histogram For a given data attribute, if this attribute is discrete, the frequency that each value occurs can be computed. If the attribute is continuous, different ranges of values can be grouped in order to enable the computation of these frequencies. A histogram is the visualization of these frequencies as a bar chart. It represents the data distribution with regards to one attribute. In the case of continuous attributes, the decision of how many groups of ranges should be considered is not trivial, as some characteristics of the data might be missed by simply dividing that range into an arbitrary number of equally sized groups.

Scatterplot It is a 2-dimensional diagram used to show the joint variation of two variables, where symbols (points or glyphs) represent data examples. Scatterplots are useful to help determine relations between attributes represented by the x and y axes. Attributes are said to be correlated if there are some dependency between them or if one influences the other in some way. If the inspection of a scatterplot suggests correlation, regression could be used to create a model of this correlation. One can also visually notice how scattered or clustered together the points representing the examples are, as well as observe patterns in the data distribution. Extensions of this basic visualization are proposed by [Chen *et al.* \(2014\)](#) and [Mayorga e Gleicher \(2013\)](#).

3.1 Perception and visual variables

Before going into specific and more elaborated visualization techniques, this section presents the visual variables at our disposal in order to convey information. The main entity in a visualization is usually a mark, or glyph. A glyph is a graphical element such as a circle or a square that can represent a data example. In order to convey information through these marks, their spatial arrangement and other factors have to be used. For instance, the position or color of the marks can encode information about the data if mapped to data attributes.

According to [Ward *et al.* \(2010\)](#), there are eight visual variables that can be used in the mapping process from data to visual marks:

Position This is the most important variable, since it is the easiest to be perceived. It is important that the data is considerably spread on the visualization. To illustrate why this is important, consider the extreme case where all the points end up being mapped to the same position.

In this case no information will be communicated. Which attributes should be mapped to position and which scale should be used are important factors to consider in order to make good use of this visual variable.

Size The size of each mark can be used to map some attribute. However it should be noted that our perception of area is not as accurate as we might think. Experiments show that humans can perceive accurately only 5 different sizes and that linear differences in area are much more difficult to be noticed than relative differences.

Shape The marks themselves can assume different shapes in order to convey information about some discrete data attribute. When using this variable it is important to choose shapes as different as possible from one another. This can be very ineffective if there are too many different values to be mapped (too many different glyphs). The user would be overwhelmed with information and therefore it would be harder to identify any sort of pattern.

Brightness An attribute can be mapped to the brightness of the mark representing the data example. It can be used to represent a continuous attribute or, through the use of a limited brightness scale, a discrete attribute.

Color While the brightness of a mark is just how light or dark the mark is, color involves also hue and saturation. Hue represents what we usually call color, that is, the dominant wavelength from the visual spectrum. Saturation is how pure the color is. If color is used to represent discrete attributes, colors as different as possible from each other should be used. If color maps a continuous attribute then an appropriate color scale should be chosen. Some examples are the rainbow scale (all hues from the visual spectrum), blue to cyan and green to red.

Orientation The effective use of this variable is only possible if the mark used clearly has a major axis, so that its orientation can be easily perceived. For instance, a circle does not change if you change its orientation, and a square looks the same after only 45 degrees of rotation. An elongated triangular mark, though, could effectively convey information by having its orientation changed.

Texture Texture can be considered a mixture of shape, size, color and brightness. Texture can map discrete attributes, but not many different ones should be used, since it might be difficult to distinguish between them.

Motion This variable is related to animation. Motion is the variation through time of some of the variables previously mentioned. For instance, the marks' positions could vary through time, so that the user is able so perceive not only similar behaviors, but also outliers (marks that do not change position in the same manner as the rest of the group). Other options would be to make marks lighter or darker, smaller or larger, etc.

It should be noted that the use of too many of these visual variables at once can be detrimental to the visualization's effectiveness, if each variable maps to a different data attribute. It is not recommended to use more than 5 or 6 visual variables at a time. If the visualization becomes too complex, it will be difficult to explore the data or identify what information is being conveyed. When possible, it is better to be redundant and map the same attribute to more than one visual variable.

There is a natural limitation in the human visual system to 2D and 3D displays (though usually 3D is just a 2D projection of 3D entities). Thus, when visualizing large volumes of data, overlapping is bound to happen, and important information might remain hidden from the viewer. In order to avoid this problem, clustering, sampling and filtering techniques can be employed (VELLIDO *et al.*, 2011; MAYORGA; GLEICHER, 2013).

3.2 Techniques for multivariate data

In Information Visualization, the data has to be transformed somehow in order to be rendered in a 2-dimensional display. With 1-dimensional, or 2-dimensional data, simple visualization methods can be used, like histograms and scatterplots. 3-dimensional data can be visualized through a 3-dimensional scatterplot, though in this case it would be very important to have a dynamic visualization where the user is able to change the visualization's point of view.

A much greater challenge is the visualization of data comprised of more dimensions (referred to as *multivariate* data, when there are some dependency between attributes), which is the most common type of real life datasets.

A way to deal with the dimensionality problem is to employ some dimensionality reduction technique prior to visualization. The methods mentioned in [subsection 2.3.1](#) can be used to this end, but there are many techniques able to represent multivariate data. In this section some of these techniques are presented.

It should be noted that the glyphs used in these visualizations do not necessarily have to be simple symbols or geometric primitives like circles or squares. If the data examples being visualized are images or text, for instance, the glyphs can be replaced with a miniature representation of the image or the text.

3.2.1 Scatterplot matrix

In this type of visualization a scatterplot is created for each pair of data attributes, so that the user can look for correlations in any of the pairs. The possible overdraw problem present in scatterplots is also present here. Also, the data might present patterns in higher dimensions that would not be apparent in this visualization (CHAN, 2006). As the diagonal of the scatterplot matrix represents the corresponding attribute paired with itself, to draw these scatterplots would

be meaningless. In Figure 2, this diagonal was used to present a histogram of each attribute.

Figure 2 – Scatterplot matrix

Source: Chan (2006).

3.2.2 *t-SNE*

In t-Distributed Stochastic Neighbor Embedding (t-SNE) (MAATEN; HINTON, 2008) dimensionality of the data is reduced to 2 or 3 so that the data can be visualized in a scatterplot. In order to accomplish this, t-SNE starts by converting the distance between points in the high dimensional space to conditional probabilities. A matrix of pairwise probabilities is created for all pairs of examples. Then a similar matrix is created in the low dimensional space. An optimization method is then used to make these pairwise probabilities as similar as possible. This method produces satisfactory visualizations even for very high dimensional data, as it keeps dissimilar examples distant and similar examples close together.

Figure 3 – t-SNE applied to the MNIST dataset

Source: Maaten e Hinton (2008).

Figure 3 shows t-SNE applied to 6,000 examples from the MNIST¹ dataset, which is a collection of handwritten digits represented by 28 by 28 matrices of grayscale values. Each different class of digit (0-9) is represented in the visualization by a different combination of glyph and color. This class information was not used in the process of creating the visualization. It is noticeable that there is a clear separation between most clusters.

3.2.3 RadViz

RadViz, which stands for Radial Visualization, was originally created to allow data visualization in 2D space, so that patterns in the data, such as clusters, could be revealed (HOFFMAN *et al.*, 1997). It is reasonable to expect these patterns to occur, since similar examples tend to be close together on the projection.

Even though RadViz is not a new technique, it still motivates many recent works (ONO *et al.*, 2015; RUBIO-SÁNCHEZ *et al.*, 2016). In this subsection RadViz is presented in detail, since it constitutes one of the basis for the experiments carried out. The results are discussed in chapter 6.

RadViz maps n -dimensional data to a plane. This technique is based on Hooke's Law, from physics. Each dimension becomes an anchor on the perimeter of a circle. Data examples are positioned inside this circle as if they were attached to n springs, represented by each anchor. The stiffness of each spring is calculated according to the attribute value for that dimension. The attribute values must be normalized before the calculation. The example's position is given by the point where the forces exerted by the springs achieve equilibrium. Equation 3.1 shows this calculation where v_{ij} is attribute j of example i and $\vec{S}_j$ is the position of anchor j .

$$\vec{x}_i = \frac{\sum_{j=0}^n \vec{S}_j \cdot v_{ij}}{\sum_{j=0}^n v_{ij}} \quad (3.1)$$

Figure 4 exemplifies the RadViz technique. It shows four data examples: A, B, C and D. Each example has four attributes. Example B has the value 1 for attribute 4 and 0 for the others, so it is completely attracted to the anchor corresponding to attribute 4.

A problem of this visualization technique is its ambiguity, since different data examples can be mapped to the same point on the projection, causing overlapping. This can be observed on Figure 4, where examples D and E are both on the center. This happens because example E is equally attracted to all anchors and example D is equally attracted to both opposing horizontal anchors (attributes 2 and 4).

Another deficiency of RadViz is that the anchors' positions are crucial to achieve a good visualization. The best arrangement, as shown by Ankerst *et al.* (1998), must be one that places similar dimensions as close as possible to each other. This problem is similar to the Traveling

¹ <http://yann.lecun.com/exdb/mnist/index.html>

Figure 4 – Basic RadViz illustration

Source: Elaborated by the author.

Salesman Problem (TSP), and therefore is NP-complete. An implementation of this technique can use some heuristic to solve the TSP, but regardless of having this automated feature, the visualization should be interactive and allow the user to manually reorder the anchors' positions.

3.2.4 Parallel Coordinates Visualization

Unlike RadViz, in Parallel Coordinates Visualization there is no ambiguity, since different examples are never represented the same way. In this type of visualization, the n data dimensions are mapped to parallel axes. A polyline representing a data example passes through each of these axes at the height corresponding to the value of each attribute. With this method one can easily observe which is the maximum and minimum value of each attribute, since usually the axes representing the dimensions are labeled with this information at each end. This makes it easy to observe the true value of each attribute for different examples, provided that there is a selection mechanism and not much overlapping. One of the deficiencies of Parallel Coordinates Visualization is the difficulty in manipulating the visualization. If one wishes to select examples that have the values of each of the n dimensions lying between a given interval, it is necessary to select this interval in each of the n axes. This visualization technique is also used in the experiments reported in [chapter 6](#).

[Figure 5](#) is an example of Parallel Coordinates Visualization applied to the Iris² dataset. Some examples having low values for the third attribute (*petal length*) have been selected and are shown in red.

3.2.5 Other techniques

Many other techniques for multivariate data visualization are surveyed by [Chan \(2006\)](#). Some popular ones are Isomap ([TENENBAUM et al., 2000](#)), Sammon Mapping ([SAMMON,](#)

² <http://archive.ics.uci.edu/ml/datasets/Iris>

Figure 5 – Example of Parallel Coordinates Visualization

Source: Ward *et al.* (2010).

1969), Locally Linear Embedding (ROWEIS; SAUL, 2000) and Star Coordinates (RUBIO-SÁNCHEZ *et al.*, 2016).

3.3 Final considerations

In this chapter the definition and goals of Information Visualization were presented. Basic visualization techniques were described as well as more advanced ones, specifically for multivariate data. The visual variables that the human visual system can perceive were also discussed, as they are used in order to convey information in visualization.

THE INTERSECTION OF MACHINE LEARNING AND VISUALIZATION

The data generated every day from areas as diverse as engineering, medicine, business, social media, etc. as well as our ability to store it, has been increasing very rapidly. On the other hand, our ability to make use of this data has not kept pace. Data Mining and ML are very powerful tools to tackle this information overload problem, but many times it is not possible to rely only in automated processes, and user involvement is necessary. This gives rise to the idea of bringing the user to a central and active position in the data analysis process. If the user can visualize and supervise the process in every step of the way, his or her knowledge can be leveraged in order to acquire the most value from the available data.

As mentioned in [chapter 1](#), the field of Visual Analytics establishes this inclusion of the user in every step of the analysis. Visual Analytics tools provide visual representations, model-based analysis and user interaction ([SUN *et al.*, 2013](#)).

This integration of automated analysis by ML processes and visualization can take many forms. For instance, ML could be used to aggregate examples or reduce data dimensionality before visualization. This interconnection is actually two-way, because just as ML can be used to improve or even enable visualization, visualization techniques can be used to assist the ML process. For instance, in contrast with the previous example, visualization can assist the user in a clustering task, by allowing the user to identify dense and disjunct groups of examples. The task of dimensionality reduction can also be assisted by visualization. For instance, observing some form of correlation between two attributes in a scatterplot is evidence for redundant information that could be discarded.

The next sections present some of the ways ML and Visualization are used together, based on different characteristics. Many examples are directly linked to Visual Analytics. Some of the works mentioned have characteristics that allow them to be present in more than one section.

4.1 Visualization in data preprocessing

As mentioned in [chapter 2](#) data preprocessing is often a necessary step prior to the employment of automated learning algorithms. Visualization can be an important tool in this process.

The field of Visual Data Mining (VDM) focuses more on the data exploration and preparation steps. For instance, ML algorithms will be more effective if the data is free from outliers and, in case of clustering tasks, if the number of clusters are previously known. VDM can be used in such cases to provide the user with visualizations that assist in outlier detection and estimation of number of clusters ([ROSSI, 2006](#); [IM et al., 2013](#)).

The data resulted from dimensionality reduction or feature extraction can also be pre-evaluated through visualization. An example can be found on ([MAATEN, 2014](#)), where after generating features from a set of images, a visualization is made by using t-SNE. The visualization can then be used to check if the resulting clusters actually preserve the target characteristics the feature extraction process was aiming to capture.

The feature selection process itself can be assisted through visualization as well. [Krause et al. \(2014\)](#) present INFUSE, which is a Visual Analytics tool that helps the user interactively extract features from data based on their predictive power. The features selected by different algorithms are fed to a series of classification algorithms. After visually evaluating the different combinations of *feature* $\times$ *classification algorithm*, the user is able to interactively build a model by using a custom feature set and a specific classification algorithm. [Figure 6](#) shows one part of the system: the scatterplot view, where features are viewed as circular glyphs. The glyph's color encode information about the feature's type. Information about the feature's rank is conveyed by the inward-filling black bars contained in each glyph. Other views allow the user to select features, change the scatterplot's axes and visualize classification performance.

Figure 6 – Infuse - scatterplot view

Source: Adapted from [Krause et al. \(2014\)](#).

4.2 Visualization assisting model creation

Predictive ML tasks are, often times, an iterative process. Visualization can be used in the intermediate steps to allow the user to steer the process towards a better predictive model. One example of visualization being used in such way, is when it is used in active learning. This use is discussed further in this section.

Another example of the use of ML in model creation is given by [Talbot *et al.* \(2009\)](#). The system, called EnsembleMatrix, uses confusion matrix visualizations. The matrices corresponding to different classifiers are shown. The user is then able to linearly combine these classifiers as to obtain a new model with better performance than any individual one. The confusion matrix of the current ensemble is also displayed. [Figure 7](#) gives an idea of how EnsembleMatrix works, with individual visualizations of each classifier on the right and the visualization corresponding to the ensemble on the left. This work is further discussed in [section 4.3](#) (on visualization to explore classification results).

Figure 7 – EnsembleMatrix interface

Source: [Talbot *et al.* \(2009\)](#).

4.2.1 Visualization in active learning

Active learning is a type of supervised learning where, at first, only a subset of examples are labeled. During training the user is asked by the system to manually label some examples. These examples are then incorporated in the training set.

Different strategies can be employed by the system in order to choose which examples the user should be queried about. Usually these examples are the ones that the model is most unsure about and that would reduce the most the future prediction error ([AODHA *et al.*, 2014](#)). In domains where labeling the instances is very costly, active learning can make the model training more feasible.

Visualization can be used in this context to provide the user with an interface to select not yet classified (or misclassified) examples. Then the user can decide whether or not to classify

(or reclassify) them.

An implementation of this idea is given by [Seifert e Granitzer \(2010\)](#). In this approach, the authors use RadViz adapted to visualize *a posteriori* classification probabilities. The examples can be positioned inside the RadViz circle based on the probability of belonging to a given class. In this scenario the anchors no longer represent dimensions; instead they represent each possible class the examples can belong to. Therefore, the points representing the data tend to group close to the anchors that represent the class with the highest probability, according to the previously performed classification. If the classification was reasonably accurate, the examples of a given class will be grouped closed to the anchor representing that class. Based on the visualization, two different picking strategies are simulated in many scenarios (involving different classification algorithms and different datasets). The results are compared to other classical picking strategies that do not involve visualization, such as uncertainty based sampling and random sampling. Seifert and Granitzer’s approach outperforms the classical strategies in most cases.

For the task of image categorization [Babae et al. \(2015\)](#) proposes a new method for visually picking examples in active learning, called First Certain Wrong Labeled (FCWL). The method consists in showing the user images ranked by classification certainty. As shown on [Figure 8](#), the visual interface presents columns of thumbnails of examples in order of classification certainty, where each column is a different category. The user then picks the first missclassified image to reclassify. The interface also plots the train and test accuracies as a function of training iterations. Results are demonstrated by applying several learning algorithms together with several other picking strategies (the data used are Synthetic Aperture Radar (SAR) images). Results show that the proposed method is better than other strategies when there are more than a certain number of examples.

Figure 8 – FCWL interface

Source: [Babae et al. \(2015\)](#).

[Höferlin et al. \(2012\)](#) proposes an *inter-active* learning system applied to video analysis. While training, the quality of the current classifier can be visualized through a cascaded scatterplot visualization, and visual interfaces for data selection and labeling are provided. The results are evaluated by the comparison of classification performance with other active-learning

sampling strategies such as random sampling and uncertainty based. It is show that the achieved classification performance is as good as the ones achieved by other methods, but using the proposed system less effort is required (less examples are labeled and the whole process takes less time).

4.3 Visualization to explore classification results

It is a common practice to analyse results from ML tasks through visualization. Effective static visualizations can be used for the purpose of performance evaluation and comparison between classifiers. For instance, bar charts can be used to compare classifiers with respect to some metric, such as accuracy. Line graphs can be used to show the change in some performance measure as some other variable changes (lines for different classifiers can be used to show how the error rate changes as number of examples increase, for instance). Another standard static visualization that helps evaluate binary classification error rates are ROC curves ([WITTEN *et al.*, 2011](#)). With an x axis representing the true positive rate (same as recall) and the y axis representing the false positive rate, a ROC curve of a classifier can be plotted by varying a parameter of the model, such as bias or threshold. This visualization, besides presenting information related to performance, can also be useful in order to compare different classifiers. For instance, if a classifier is always better than another, its ROC curve would be closer to the upper left. A modification of ROC curves, called cost curves, is proposed by [Drummond e Holte \(2006\)](#). Additional information, such as the classifier's performance given a specific misclassification cost, can be visualized through cost curves.

Going beyond basic visualizations, many have proposed more sophisticated, often times interactive, visualizations for exploration of classification results. This section discusses some of these works.

As the insights generated by visualizing classification results can frequently be used to improve the classification model, some systems support the user to rebuild the model based on the visualization. This can be done in different ways. For instance, the system can support active learning or creation of ensembles of classifiers. Therefore, it is worth noting that the systems with this *model creation* aspect that are mentioned in this section could also be accounted for in [section 4.2](#).

One such system is the already mentioned EnsembleMatrix ([TALBOT *et al.*, 2009](#)). The authors use heatmap visualizations, each representing a confusion matrix that comes from a classification task that have been previously undertaken. As seen on [Figure 9](#), the heatmap version of a confusion matrix (on the right), despite not containing precise information like the traditional version (on the left), is easy to interpret. It is noticeable that classes 8 and 3 are confused with one another and that class 5 is misclassified as many different classes.

Another work that, besides providing methods for visualization of classification results

Figure 9 – Confusion matrix comparison

Source: Adapted from Talbot *et al.* (2009).

also supports model creation, is the one proposed by Paiva *et al.* (2015). This time in the context of active learning. In this work Neighbor Joining Trees are used to visualize data examples. Neighbor Joining Trees are similarity trees: their structure is derived from the similarity between the examples, so that examples in the same branch tend to be from the same class. In these trees, the examples positioned far from the center of the tree (for instance, the examples represented by the most external leaves) are the ones that better characterize the class they belong to. It follows that the examples closer to the center of the tree are the ones whose features are not really characteristic of any class, or that may appear to belong to more than one class. Two different ways are provided to visually analyze classification results based on the coloring of the nodes:

- The user can compare two versions of the tree where the nodes' colors represent each of the different classes contained in the data. One tree is colored according to the examples' true classes and the other is colored according to the classification result;
- The classified examples are shown on the similarity tree and their colors are either green or red, indicating examples correctly classified or misclassified, respectively.

Figure 10 – Neighbor Joining Trees

Source: Paiva *et al.* (2015).

The use of these visualizations allows the analysis of different aspects of the result. For instance, if similar branches are assigned the same class, it is expected that branches

are homogeneous in terms of colors. Branches with high proportion of mixed colors should be inspected. This might mean the classifier was unable to learn the defining features of the examples represented in those branches.

Figure 10 shows two trees representing image data. The tree on the left is colored according to the examples' actual class, and the one on the right shows the misclassified examples in red. This work also supports iterative classification, where the user can immediately apply new insights to try to improve the classification.

A very complete tool for analysing classification results of multiclass data is given by Alsallakh *et al.* (2014). This visualization tool for *a posteriori* analysis of probabilistic classification data encompasses scatterplots, histograms and a novel visualization called confusion wheel. Figure 11 shows a part of a confusion wheel. Each sector shown represents a class and contains bars representing misclassified examples (false negatives and false positives). The bars' dimensions correspond to how many examples they represent. The ordering of the bars in each sector communicate how confident the classifier was on its output: bars in the inner part of the sector represent low confidence and in the outer part represent high confidence. The colors of each bar correspond to the true class of the examples represented by that bar. The chords of varying thickness connecting the sectors convey information about classification confusion between classes.

Figure 11 – Confusion wheel

Source: Alsallakh *et al.* (2014).

As mentioned in section 4.2, RadViz can also be used to explore probabilistic classification results. Ono *et al.* (2015) developed a system, named Concentric RadViz in order to visualize data previously submitted to multi-label classification. In Concentric RadViz, concentric circles contain the anchors representing the classes of each classification task through which the multi-label data has been submitted. The user can rotate the circles and align anchors. By doing this it is possible to filter and explore different aspects of the data.

4.4 ML to enable or improve visualization

Previous sections discussed the use of Visualization to improve ML processes, but ML can also be used in order to assist in the creation of more effective visualizations.

For instance, for large volumes of data, a clustering algorithm can be applied prior to visualization, so that homogeneous groups of examples could be replaced by representative ones. This would enable the creation of visualizations with less clutter and overlapping (ROSSI, 2006).

A very successful visualization tool based on ML is the Self-Organizing Map (SOM) (KOHONEN, 1990). SOM is a Neural Network algorithm based on unsupervised learning. The network has only an input layer and an output layer. The SOM's neurons are usually arranged in a 2D grid and each neuron is represented by a vector of the same dimensionality as the data. When a data example (also a vector) is presented to the SOM during training, the neuron whose vector is closest to the example's is chosen. This neuron's vector, and its neighbors', are updated to better match the example's vector. As a result, the SOM acts as a clustering algorithm. The 2D arrangement of the neurons makes it easy to be visualized, also solving the problem of superposition. There are several kinds of visualizations based on the SOM (VESANTO, 1999). One example is the Image Feature Map (IFM), proposed by Goncalves *et al.* (2008), in the context of remotely sensed images. In this case, the input data are pixel windows sampled from the remotely sensed image. As each neuron's vector is of the same dimensionality as these pixel windows, it is possible to generate an image of the neural grid of the SOM, as seen on Figure 12 (right). The image on the left of Figure 12 is the one used to train the SOM represented by the IFM on the right.

Figure 12 – IFM - example of SOM-based visualization

Source: Goncalves *et al.* (2008).

An interesting example of the use of ML to enable visualization is a modification of the previously mentioned t-SNE (subsection 3.2.2), called parametric t-SNE (MAATEN, 2009). This technique uses a deep Neural Network to map high-dimensional data to low-dimension space while preserving the data's local structure, similarly to regular t-SNE. The main difference is that on this approach the transformation is parameterized (by means of the weights on the trained Neural Network). The generalization error is shown to be minimal when compared to other dimensionality reduction techniques, such as PCA. This means that, after training, new

data examples could be promptly mapped to the low-dimensional space in order to be visualized, which is not possible with regular t-SNE.

Another example comes from the field of Scientific Visualization, where the original data usually has spatial meaning (usually 3D data from fields such as biology, meteorology, engineering, etc.)(KEIM *et al.*, 2010). Ma (2007) proposes a system for high-dimensional volume visualization. This sort of visualization requires a transfer function, that usually maps a value associated to a 3D point (its density, for instance), to visual properties, such as color and opacity. The creation of an appropriate transfer function can be quite challenging to do by hand, so the proposed system uses ML to create such function, with a simple input from the user.

Figure 13 – Intelligent, high-dimensional volume classification

Source: Ma (2007).

In order to render the different parts of the volume differently, the user is provided with an interface to paint the areas of interest on a basic visualization of the volume (top of the Interface box on Figure 13). These painted regions are used as training data in a Neural Network so that the whole volume can be classified. The classification assigns to each voxel the likelihood that the voxel is part of the area of interest. On the final visualization this likelihood is mapped to opacity.

4.5 Visualization to provide insight into black-box ML algorithms

Traditional ML solutions are often black-boxes, that is, even if the recommended solution is accurate, there is no explanation on why.

If an ML system is able to provide understandable descriptions and labels of its outputs and internal states, domain knowledge and intuition from the user can be inserted in the process of pattern detection and classification (PAIVA *et al.*, 2015).

Insights into black-box algorithms, such as Neural Networks, can be interesting in their own right, just by showing the algorithm's inner workings. But these insights can also be beneficial in practical ways. One example is the increase in confidence that happens when

the user understands why certain output was returned by the ML model. Another example is that, by understanding the model's inner workings, sometimes the user can find deficiencies or specific points that should be changed in order to improve performance. In the context of image classification with Convolutional Neural Networks (CNNs), this is exactly what happened when, in 2013, Zeiler and Fergus created a novel way to visualize what was learned by the network (ZEILER; FERGUS, 2014). Their deconvolutional method allowed the feature activations to be projected back into the input image, thus making these learned features visible. These visualizations suggested architectural changes to the ML model allowing them to outperform the current state of the art on the ImageNet classification benchmark.

Recently there has been some increased interest in understanding and representing what a CNN has learned (SAMEK *et al.*, 2016; RADFORD *et al.*, 2015; MAHENDRAN; VEDALDI, 2016). As an example that builds on the work of Zeiler e Fergus (2014) and many others, there is the software tool developed by Yosinski *et al.* (2015), which brings together many visualization methods for CNNs, and contributes with a new one. For instance, the user can pick a certain layer from the network and then pick an individual neuron through an overview of all neurons. For a given input (image or video), the user can see which parts of the unit activates, if any. Figure 14 shows this visualization for one particular neuron and different inputs. The bottom row shows the visualization of a given neuron when the input is the corresponding image on the top row. In the case illustrated here one can see that the neuron responds to faces, even though it is reported that there was no "face" class when training the network. The contribution made by Yosinski *et al.* (2015) is illustrated in Figure 15. After a neuron is picked, the tool synthesizes multiple images to illustrate what the neuron is responsible for recognizing. In this case, the image shows features that resemble a cat (fur and catlike ears can be noticed).

Figure 14 – Visualization of Deep Convolutional Neural Networks

Source: Adapted from Yosinski *et al.* (2015).

SVM is known for being a robust algorithm, but it is also regarded as a black-box modeling method. Migut *et al.* (2015) argues that an intuitive way to understand black-box

Figure 15 – Synthetic images for CNN Visualization

Source: Adapted from Yosinski *et al.* (2015).

classification models such as SVM is through the use of scatterplots showing the classifier's decision boundaries. With this method, one can visualize the scatterplot of any pair of features and observe how the model separated the examples. Figure 16 shows the decision boundary for two pairs of features from a 10-dimensional dataset as classified by SVM. The misclassified examples are marked with a green square.

Another approach is the extraction of rules from SVMs. Martens *et al.* (2008) covers many rule extraction methods. Some methods consist in inducing a decision tree that can be easily visualized and interpreted.

Figure 16 – Decision boundaries for SVM classifier

Source: Migut *et al.* (2015).

4.5.1 Prospector and the used of partial dependence plots

The next Visual Analytics system to be discussed is based on the concept of partial dependence.

4.5.1.1 Partial Dependence Plots

The concept of partial dependence consists in computing the dependence of the target variable (class probability or a real value, for instance) on an attribute or set of attributes (the values for the other attributes could be the average for the whole dataset).

A Partial Dependence Plot (PDP) is a graph presenting the effect that changes to a given attribute cause on the output of a particular model.

Figure 17 – Partial Dependence Plot

Source: [Hastie et al. \(2009\)](#).

Using the California Housing dataset ([PACE; BARRY, 1997](#)), the PDP on [Figure 17](#) shows the partial dependence on house value with respect to median income of those living in certain California neighborhoods. The plot shows that as the median income on the area increases, so does the house prices on the neighborhood.

As reported on ([KRAUSE et al., 2016](#)), partial dependence for a given attribute a is given by

$$pdp_a(v) = \frac{1}{N} \sum_i^N pred(x_i) \text{ with } x_{ia} = v \quad (4.1)$$

where N is the number rows in the input matrix x (only the attributes of data examples), $pred$ is the prediction function of the model, which takes a row from x (an attribute vector) and outputs a score, and a is an attribute. The result is the average outcome over all input rows while changing the value of attribute a to the input value v for each row x_i . The PDP can be achieved by plotting the range of values of v on the x axis while plotting the respective average outcome on the y axis. With the plot it is possible to observe the overall influence of attribute a on the prediction score.

4.5.1.2 Prospector

Krause *et al.* (2016) present a Visual Analytics system named *Prospector*. This tool takes a previously constructed single class predictive model and helps data scientists understand the impact of each attribute on the classification.

The user can also inspect specific instances and even tweak attribute values in order to check what impact this would have in the classification. Among the visualizations used are PDPs. Average and standard deviation are also shown in the plot and the attribute's distribution is shown in a histogram. A novel visualization (partial dependence bar) is presented and used to provide interactivity while inspecting individual examples. The selection of examples is made through a visualization that categorizes the examples in meaningful ways to the user. Figure 18 shows the visualizations used in Prospector. The PDP, histogram and partial dependence bar are on the left. On the right is shown the visualization used to select data examples. A limitation of this work is that the tool provided assumes the data to be composed of independent (orthogonal) attributes, so correlations and influences between attributes are not taken into account.

Figure 18 – Analysing partial dependence with Prospector

Source: Krause *et al.* (2016).

4.6 Summary of reviewed works

Following the previously presented characterization of the uses of ML alongside with visualization, Table 2 shows the main reviewed works in terms of these characteristics.

4.7 Final considerations

This chapter discussed and exemplified the diversity of applications of visualization in ML tasks. It is noticeable that the intersection of the two fields is vast and good results were

Table 2 – Reviewed works

Works	Application of Visualization				
	Data preprocessing visualization	Assist model creation	Classification results exploration or performance analysis	Insight into black-box algorithms	ML to enable or improve visualization
(ALSALLAKH <i>et al.</i> , 2014)			✓	✓	
(BABAEI <i>et al.</i> , 2015)		✓			
(DRUMMOND; HOLTE, 2006)			✓		
(GARG <i>et al.</i> , 2008)		✓			
(GONCALVES <i>et al.</i> , 2008)	✓				✓
(HÖFERLIN <i>et al.</i> , 2012)		✓	✓	✓	
(KRAUSE <i>et al.</i> , 2014)	✓	✓			
(KRAUSE <i>et al.</i> , 2016)			✓	✓	
(MA, 2007)					✓
(MIGUT <i>et al.</i> , 2015)		✓	✓		✓
(MAATEN, 2014)	✓				
(ONO <i>et al.</i> , 2015)			✓		
(PAIVA <i>et al.</i> , 2015)		✓	✓		
(SEIFERT; GRANITZER, 2010)		✓	✓		
(TALBOT <i>et al.</i> , 2009)		✓	✓		
(YOSINSKI <i>et al.</i> , 2015)			✓	✓	

Source: Elaborated by the author.

achieved in several areas. A characterization of these different applications was made in order to organize the reviewed works.

METHODS: WBVA

With the focus on flexible visual data exploration and analysis of classification results, this chapter presents a web-based system that comprises a number of ML algorithms, visualization techniques, and means of interaction. This system will be addressed as WBVA¹ for brevity. IVEDCS

WBVA is implemented in JavaScript by using Data-Driven Documents (D3)² for the visualization, which is a well known JavaScript library designed for this purpose. The system's back-end consists of an R³ server, in which ML related tasks are performed, such as model training and data classification. The ML classification algorithms available to the user come from the *caret*⁴ R package. The system does not make any assumption about which classifier was used, but the use of a classifier able to output probabilistic values is recommended, because it better justifies the use of RadViz, as will be explained later. That being said, all the algorithms available are probabilistic ones.

The system's objective is to support the tasks described on objectives 2 and 3, on [section 1.3](#), that is, allow the user to:

- Get a sense of class separation within the dataset;
- Get a sense of how particular attributes correlate to the examples' true classes;
- Identify examples missclassified with high confidence or correctly classified with low confidence;
- Inspect if by changing the value of an attribute for an example or set of examples, the result of the classification output changes;

¹ The system can be accessed at: <http://igorbc.ocpu.io/wbva>

² Data-Driven Documents: <https://d3js.org/>

³ The R Project for Statistical Computing: <https://www.r-project.org>

⁴ The *caret* package: <http://caret.r-forge.r-project.org>

- Identify attributes that do not impact significantly the classification result, regardless of their values.

The next four sections present the main views of the system and the reasoning behind them. On [section 5.1](#) the modified version of RadViz is shown, which is the system's main visualization, and it is also explained how the examples' positions are calculated within the visualization. On [section 5.2](#) the alternative color scheme that is available when using the modified RadViz is presented. On [section 5.3](#) it is briefly discussed the role of Parallel Coordinates visualization in the system. On [section 5.4](#) the multiclass Partial Dependence Plots used in the system is presented.

On [section 5.5](#) the system as a whole is presented and on [chapter 6](#) detailed usage scenarios are shown and discussed.

5.1 Dual RadViz: A modification to the RadViz technique

WBVA's main visualization is a modified version of RadViz which allows the user to smoothly transition between traditional visualization of the data based on its attributes and visualization of the classification results based on the probability of each example belonging to each class. This modified version is addressed here as Dual RadViz ([CORRÊA; CARVALHO, 2016](#)).

Dual RadViz consists of two concentric circles containing glyphs representing the data examples. The inner circle is the dimension group (DG), and contains the anchors representing each data attribute. The outer circle is the class group (CG), and contains the anchors representing each possible class. All examples are positioned inside the inner circle.

In order to calculate the position of each point representing the data examples, two modifications were performed in the traditional calculation:

- When being calculated, the positions of the outer circle's anchors (class group) are projected onto the inner circle (dimension group), as if the outer circle had the same radius as the inner circle. As a result, the examples are always positioned inside the inner circle;
- A contribution factor c , with value varying between 0 and 1, is bound to each group (DG and CG).

Therefore, if the contribution factor value is 1 for both groups, all anchors have the same influence on the positioning of the data (50%). If the factor value is 1 for one group, for example, DG, and 0 for the other, the visualization would consider only the dimension group (the class

group would have no consequence, since this configuration is equivalent of DG having 100% of influence). This new calculation is presented on Equation 5.1.

$$\vec{x}_i = \frac{\sum_{j=0}^n \vec{SD}_j \cdot v_{ij} \cdot c_{DG} + \sum_{k=0}^m \vec{SC}_k \cdot p_{ik} \cdot c_{CG}}{\sum_{j=0}^n v_{ij} \cdot c_{DG} + c_{CG}} \quad (5.1)$$

Where n is the total number of dimensions, $\vec{SD}_j$ is the position of anchor j from DG, m is the total number of possible classes, $\vec{SC}_k$ is the position of anchor k from CG, v_{ij} is the value for dimension j from example i , p_{ik} is the probability of example i belonging to class k , and c_{DG} and c_{CG} are the dimension group and class group contribution factors, respectively.

Figure 19 – Slider position and influence of each anchor group

Source: Elaborated by the author.

The user is provided with a slider in order to control the contribution factor. The slider's position is mapped to both coefficients as to ensure that, always, at least one of them is equal to 1. Figure 19 shows the coefficient values for c_{CG} and c_{DG} , as well as the percentage of influence CG and DG have on the projection as the slider's position varies (x axis).

Figure 20 – Dual RadViz with examples from the Iris Dataset

Source: Elaborated by the author.

(a) class legend, (b) slider to determine CG's and DG's relative influence, (c) Class Group, (d) Dimension Group.

Figure 20 is an illustrative example of Dual RadViz using the Iris dataset. In this example the the anchor's positions were set manually with no particular intent. Detailed usage examples are shown on chapter 6.

5.2 The *Confusion Colors* color scheme

If the spatial positioning of examples within the class-based RadViz is unclear to determine whether some instances were classified correctly or incorrectly, a new color scheme is available, called *confusion colors*. With this color scheme the examples' colors cease to represent their classes and start to illustrate the classification results with respect to the queried class. The color scheme chosen follows the pattern used in the confusion wheel visualization proposed by Alsallakh *et al.* (2014) and is as follows:

- **Green** - TP (True Positives)
- **Red** - FN (False Negatives)
- **Gray** - TN (True Negatives)
- **Yellow** - FP (False Positives)

Figure 21 – Illustrative example of *confusion colors*

Source: Elaborated by the author.

To activate this color scheme the user interacts with the label of one particular class by clicking. Figure 21 is an illustrative example, again using the Iris dataset, where *confusion colors* are applied with respect to the Iris-setosa class. In this case all data examples were correctly classified, since there are only green (TP) and gray (TN) circles within the visualization. Figure 21 also shows that when *confusion colors* are active the appropriate legend is displayed on the top right.

5.3 Combination of Parallel Coordinates Visualization and RadViz

As mentioned in [subsection 3.2.4](#) (about Parallel Coordinates Visualization), the selection of examples in this visualization is a tedious process. In RadViz it is trivial to select a subset of examples positioned inside the circle by using the mouse to draw shapes around the points of interest. The technique named *brushing and linking* ([DOLEISCH, 2004](#)) allows the combination of the two visualizations by making the selection made on the RadViz view be reflected on the Parallel Coordinates Visualization (the points highlighted on the RadViz will correspond to polylines highlighted on the Parallel Coordinates Visualization). The combination of these two types of visualization is done by [Bertini et al. \(2005\)](#), where clustering techniques are also used to minimize cluttering and overlapping in the Parallel Coordinates Visualization.

WBVA also employs this combination, leveraging this ease of selection on the RadViz. The presence of Parallel Coordinates Visualization on WBVA also provides an unambiguous alternative to RadViz.

5.4 Multiclass Partial Dependence Plots as means of interaction

On [subsubsection 4.5.1.1](#) it is mentioned that PDPs can help visualize the impact certain attributes have on the classification result. On WBVA a multiclass version of PDPs is used for this purpose. After training a model it is possible to visually and interactively analyze the impact that any data attribute has on the behavior of that particular model through these multiclass PDPs.

[Equation 4.1](#) describes the computation of single-class PDPs, and as such the result is the average probability of examples belonging or not belonging to the one class. In a multiclass scenario the following similar computation is performed in order to obtain the PDP.

$$multipdp_a(v, C) = [avgpred(x, c_1), \dots, avgpred(x, c_n)] \text{ with } x_{ia} = v \quad (5.2)$$

In [Equation 5.2](#) C is the collection of classes present in the data and n , which also represents the total number of classes, is the number rows in the input matrix x . $avgpred$ is the average prediction score for the input matrix x and the particular class c , and could be written as

$$avgpred(x, c) = \frac{1}{N} \sum_i^N pred_c(x_i) \quad (5.3)$$

As the calculation of [Equation 5.2](#) is performed for every existing class, the resulting vector sums to 1.

Figure 22 – Multiclass PDP examples

Source: Elaborated by the author.

Figure 22 shows some illustrative multiclass PDP examples, where colors represent each existing class. The plots on each column are equivalent, the only difference being the ones at the top show the average probability as stacked areas and the ones at the bottom as line graphs. On both graphs on the left it is noticeable that the selected attribute on which the Partial Dependence is being calculated has a dramatic effect on the average probabilities: at very low values most examples are classified as Class 3, examples with low to mid values as Class 1, and at the higher end as Class 2. On the graphs on the right the attribute selected does not seem to matter at all, since as the values vary the probability average stays the same.

The interactive aspect refers to the user's following ability: after selecting an attribute and one or more examples, the user can vary the attribute values for the selected examples (through the use of a slider) and instantly observe the impact that this change causes on the classification result, as this will change the example's position on the RadViz. Varying attribute value can also change the examples' color if their classification changes and the color scheme is set to *confusion colors*.

5.5 WBVA environment

Figure 23 shows the main view of the system after a classification task has been performed. First, on Figure 23(a), we have the menu bar where each option reveals a side bar through which the user can perform the related actions.

The *settings* option allows the user to customize size and opacity of the circles representing the data on Dual RadViz, which can be useful depending on the amount of data and incurrence of overlap.

Through the *data* option one can load a local CSV (comma-separated values) file containing the dataset to be explored and used to train a ML classification model. The *training* option allows the user to pick the ML algorithm to be used among a selection provided by the R server. Before training the model the user can choose what percentage of the data should be used for training and what percentage should be the hold-out. This hold-out data is what will be used in the post-classification analysis within WBVA. When choosing to train the model, the appropriate amount of data (according to the percentage chosen) is sampled in a stratified manner and used to train the model.

After the training phase is completed the hold-out data is then classified using the generated model and the system is updated. This update affects the main visualization (Dual RadViz, Figure 23b) and the Parallel Coordinates visualization (Figure 23c). The automatic classification of the hold-out data also populates the bottom bar (Figure 23e) with accuracy and the Kappa coefficient, as well as a heat-map version of a multiclass confusion matrix. These traditional metrics along with the confusion matrix can help the user have an immediate meaning of the classification performance.

Figure 23d shows a panel listing the predictive attributes (or dimensions) of the data. The user can enable and disable any attribute and see the effect on the system's visualizations. If the user chooses to disable certain attributes before training, those will not be used in the induction of the model.

Figure 23 – WBVA overview

Source: Elaborated by the author.

(a) menu bar, (b) main visualization (Dual RadViz), (c) Parallel Coordinates Visualization, (d) dimensions panel, (e) status bar with classification information

Other options included on *settings* allow the user to opt to have both the Dimension Group and Class Group from Dual RadViz to be superimposed and make the opacity of each group (and it's related anchors) vary dynamically according to the slider position. This dynamic opacity variation can make more immediately clear to the user which is the most dominant aspect of the Dual RadViz, that is, if the positioning of examples represent mainly the classification results or the data itself. Figure 24 illustrate the use of these settings.

Figure 24 – Illustration of Dual RadViz settings

Source: Elaborated by the author.

(a) separated groups 80% attribute-based, (b) superimposed groups 80% attribute-based, (c) superimposed groups 100% attribute-based, (d) superimposed groups 100% class-based

On Figure 24a the Class Group is semi-transparent, since it is contributing for 20% of the examples' positioning. Figure 24b is equivalent, but the groups are superimposed. Figure 24c has superimposed groups and dynamic opacity as well, but as the slider implies an entirely attribute-based visualization, the Class Group is completely transparent. Figure 24d is the opposite, since the visualization is entirely class-based. In some cases the use of superimposition might seem to make the visualization clearer, but this might not be the case if the dataset has too many classes or dimensions, since the anchors and their labels would overlap, hindering readability.

RESULTS

This chapter presents as the results some usage scenarios with the goal of showing how the tasks mentioned on [section 1.3](#) can be put into practice. The datasets and algorithms used are presented next.

6.1 Datasets

The main visualization technique being used (RadViz) suggest datasets with meaningful, human-understandable attributes. This is because information about the attributes are preserved in the visualization (as opposed to what happens on other multi-dimensional visualization techniques, such as t-SNE). Using datasets with extracted features or attributes that the user cannot interpret is possible, but not preferred, since it would configure a sort of waste of the visualization's potential. It is also desirable that the attributes are numerical, since the positioning of the examples on the visualization depend on normalized attribute values.

Two datasets from the UCI¹ repository which comply to the desired characteristics mentioned above are used in the scenarios described on [section 6.3](#). They are:

Vertebral Column dataset ² This dataset has 310 examples with 6 numerical (continuous) attributes. The values come from biomedical measurements and there are no missing values. The examples are labeled as belonging to one of three classes, each representing a diagnosis: DH (disk hernia, 60 examples), SL (spondylolisthesis, 150 examples) and NO (normal, 100 examples).

Thyroid Disease (New Thyroid) dataset ³ The New Thyroid dataset has 215 examples with 5 numerical (continuous) attributes, originating from 5 lab tests made to predict the patient's

¹ UCI datasets: <https://archive.ics.uci.edu/ml/datasets>

² Vertebral Column at UCI: <https://archive.ics.uci.edu/ml/datasets/Vertebral+Column>

³ Thyroid Disease at UCI: <https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease>

thyroid condition. The examples' class is whether a patient has hyperthyroidism (35 examples), hypothyroidism (30) or is normal (150). This dataset does not contain missing values either.

6.2 Algorithms used for training

Several classifier algorithms were used. The ones reported on are Model Averaged Neural Network (RIPLEY, 2008) and Recursive Partitioning and Regression Trees (THERNEAU; ATKINSON, 1997), all of which have R implementations which are available to the user within WBVA. These algorithms were used because of all of the 13 algorithms available on WBVA, they were the ones with highest accuracy and Kappa coefficient for the given dataset.

The data was split so that 20% was retained as hold-out data for final model evaluation. The training procedure consists in varying the algorithm's parameters and for each combination the performance of a 10-fold cross-validation procedure is averaged. The best model is then returned to be used in the classification of the hold-out data. This procedure happens automatically and the user has no control over the parameter combinations or number of folds for the cross-validation.

6.3 Usage scenarios

Figure 25 illustrates the New Thyroid dataset after the classification task was performed with the Model Averaged Neural Network algorithm. One of the system interactive features is the ability the user has to reorder the RadViz's anchors in order to achieve a clearer visualization (less overlap, clearer separation of classes, among other desirable features.). After manually reordering the dimension anchors on RadViz (visible on Figure 25a, which is 100% attribute-based RadViz, thus probability of belonging to each class is ignored on this configuration), it can be observed that the visualization shows little overlap between classes.

The visualization on Figure 25a also suggests that examples from the class *normal* have higher values for dimensions Tyroxin and T3resin, while having lower values for the remainder. Examples of the *hyperthyroidism* (*hiper_t*) class have, in general, higher Triiodothyronine and Thyroxin, placing the examples closer to the top of the visualization. The lower part of the visualization mainly shows *hypothyroidism* (*hipo_t*) examples, which have higher values for TSH_value. Also on Figure 25a, to illustrate the brushing functionality, four examples were selected. These examples lie on the border of a cluster of *normal* examples, even though three of them are from the class *hypothyroidism*.

On Figure 25b the slider has been dragged all the way to the left, making the visualization entirely class-based. As the selection is preserved, this action of smoothly shifting the type of the visualization is a way of preserving the visual context between visualizations with different

meanings. The user could want to know how these particular four examples that are clustered with *normal* ones were classified, and after moving the slider, that is the visualization arrived at. One can see that, from the *hypothyroidism* examples selected, two were misclassified, since they appear closer to the *normal* anchor. There might still be doubt about the classification result for the true *normal* example selected. On [Figure 25c](#) *confusion colors* were associated to the *hypothyroidism* class (see dark outline on the *hipo_t* legend), making clear that indeed the two *hypothyroidism* examples were misclassified (false negatives) and showing that the *normal* example was correctly classified (true negative). [Figure 25c](#) also shows that by pointing the mouse on a individual example, its detailed information is displayed for inspection. In this case, it confirms that the selected *normal* example was correctly classified with a relatively low score from the model (52% against 47%).

Regarding transparency, in a real case scenario, the two subjects with hypothyroidism might want an explanation for their misdiagnosis. The interactive PDPs on WVBA can support counterfactual explanations for this case. On [Figure 26](#) PDP on dimension T3resin can be visualized. The slider above the PDPs can be used to vary the attribute value for the selected examples (in this case, only one). By doing so, the examples on the RadViz move accordingly. If *confusion colors* are active, the example's color is also updated if its classification changes as the attribute value changes.

On [Figure 26a](#) the example in question has its original value for T3resin. On [Figure 26b](#) the slider was used to increase the T3resin value just enough so that the model classifies the example correctly. In this scenario, the subject represented by this data example could be given the following counterfactual explanation: “You were not diagnosed with hypothyroidism because your measurement for T3resin was 102. If it had been 106 or larger you would have been diagnosed with this disease.” By varying other attributes and interacting with the respective PDPs, it is possible to arrive at other counterfactual explanations.

Next, on [Figure 27](#), Recursive Partitioning and Regression Trees are used to classify the Vertebral Column dataset. As with the usage scenario with the New Thyroid dataset, all algorithms available on WBVA were used. Those whose predictive performance is being reported had the highest accuracy and Kappa coefficient.

The slider on [Figure 25a](#) was positioned so that the RadViz shows a visualization based only on predictive attribute values. Even after manually rearranging the anchors, class separation is not very clear.

On [Figure 27b](#), the slider was moved all the way to the left, so that the visualization represents the probability of the examples belonging to each class. When visualizing classification probabilities, usually many points overlap close to the anchors representing the class groups (considering that the classification was reasonably successful). In this case, a severe problem of overlapping is observed. This makes it impossible to visualize the volume of examples classified as each class, and also selecting individual examples on these areas of the projection. This

problem can be partially mitigated through the use of the slider. The slider can be used in an intermediate position still close to the left end, but not *on* the left end. By doing so, it is possible to obtain some spreading of the data, making it possible to select examples. Using the slider in such a way can be considered an alternative to introducing artificial noise to the data. This is shown on [Figure 27c](#), where it can be seen that one *normal* example (NO, in green) was misclassified as *spondylolisthesis* (SL, in orange) with high confidence. Further inspection shows that the example was classified as SL with 98% confidence. This misclassification can be considered a case of an *unknown unknown*. [Attenberg et al. \(2011\)](#) highlights the problem of *unknown unknowns* in classification. This occurs when some example or set of examples are misclassified, but the model is highly sure about this output. It is argued that discovering *unknown unknowns* is valuable information. It might mean, for instance, that important and representative examples were absent when training the model.

Further inspecting the results from the classification procedure on the Vertebral Column dataset, [Figure 28](#) shows the use of PDP for the pelvic incidence dimension. It can be seen that the average probability distribution as the value for pelvic incidence varies does not change. By varying the value from the lowest ([Figure 28a](#)) to the highest ([Figure 28b](#)), it is observed that the model does not change its classification output for any example. The irrelevance of this dimension for this classification algorithm was tested by, still within WBVC, retraining the model after excluding pelvic incidence. Accuracy and the Kappa coefficient remained the same.

In contrast with pelvic incidence, [Figure 29](#) shows PDP on an important dimension of the Vertebral Column dataset: sacral slope. On [Figure 29](#) *confusion colors* are active and the class/attribute slider is set to 60%/40%, to provide a reasonable spreading of the data while keeping examples closer to the anchors representing the classes they were assigned to. The PDP shows how increasing values for sacral slope tend to dramatically affect the model's output for examples from the classes *normal* (NO) and *disk hernia* (DH). More specifically, with low values most *normal* examples are misclassified as *disk hernia* ([Figure 29a](#)), and with high values no examples are classified as *disk hernia*. A noticeable fact about the PDP is that the changes on model output happen abruptly when varying attribute value. Even though WBVA is algorithm-agnostic, this is somewhat revealing of the inner workings of the model. In this case, we know that the model used is decision-tree-based, which can be the reason for the abrupt output changes. The overlapping previously shown on [Figure 27b](#) is also related to the nature of the classification algorithm.

Figure 25 – New Thyroid dataset classified with Model Averaged Neural Network

Source: Elaborated by the author.

Figure 26 – PDP on dimension T3resin for misclassified example

Source: Elaborated by the author.

Figure 27 – Different visualizations through the use of the slider for the Vertebral Column dataset

Source: Elaborated by the author.

(a) only DG influences RadViz, (b) only CG influences RadViz, (c) distributed influence as to provide larger spreading of the data.

Figure 28 – PDP on dimension *pelvic incidence*

Source: Elaborated by the author.

Figure 29 – PDP on dimension *sacral slope*

Source: Elaborated by the author.

CONCLUSION, LIMITATIONS, AND FUTURE WORK

This work presents a freely available web-based Visual Analytics system that supports tasks related to exploratory data visualization and ML. One of the visualizations available is Dual RadViz, which is a novel alternative for inspecting, in the same visualization, data based on its attribute values and based on the *a posteriori* result of a probabilistic classification task, and is implemented by altering the traditional RadViz algorithm. The already established advantages of using RadViz are preserved, adding the benefit of seamless transition between the two uses of RadViz. The known advantages of the combined use of Dual RadViz and Parallel Coordinates Visualization are also incorporated, increasing the level of detail possible on the exploration process. With the option of using the *confusion colors* color scheme, the analysis of the classification results is possible even when Dual RadViz falls short in attempting to provide a clear visualization.

The proposed method was evaluated in the usage scenarios discussed on [section 6.3](#). These scenarios show how, through visual data exploration, one can extract potentially useful information. The examples with the New Thyroid dataset describe how a general understanding of the data can be achieved through visualization. It is possible to observe class separation and to infer which attributes seem to correlate to specific classes of data examples. It is also shown how the interactive Partial Dependence Plots can help with the challenge of transparency by providing means to arrive at counterfactual explanations for the model's output. The usage scenario with the Vertebral Column dataset shows how examples misclassified by the model with high confidence can be identified through the visualization. In this scenario it is also argued that PDPs can help identify irrelevant dimensions and provide a glimpse of the inner workings of the ML model.

Regarding Dual RadViz, one limitation of the current implementation is the absence of an automated procedure to position the anchors in a way that maximizes the spreading of the data and class separation. The option of manual interaction with the anchors' positions is given

to the user, but due to the many possible arrangements, it is unlikely that the user is confident on the chosen arrangement's quality. The integration of techniques able to automate this procedure, such as the one discussed by [Caro et al. \(2010\)](#), could improve the usefulness of the developed system.

The use of WBVA with high volumes of data would cause a high overlapping on the radial visualization, as well as on the Parallel Coordinates visualization. Automatic clustering, as used by [Bertini et al. \(2005\)](#), could be employed to minimize this problem.

A problem inherent to this area is the evaluation of results. Visualizations are used to gain insight and uncover information from the data being visualized. Thus, the information is unknown prior to visualization. If no information about the data is known beforehand, and some new information is obtained after visualization, there is nothing to compare this new information to, in order to assess its correctness ([CHAN, 2006](#)). Nevertheless, comparative studies and tests with users could be made in order to estimate the usefulness of the methods proposed in this work. This would provide a more reliable result than the visual exploration that was carried out.

Future work could also include automated attribute ranking based on their importance for the classification output. This would enable a more assertive analysis of partial dependence, preventing the user from deeply analyzing unimportant attributes before more relevant attributes.

Besides increasing the number of classification algorithms available to the user from within WBVA, an important future work would be to expose to the user more parameter configurations for the training procedure. This would allow a more directly iterative approach to the classification process. The functionality of directly comparing different models could also be useful on this iterative search for understanding and more reliable data classification.

A paper about Dual RadViz was published at the 5th Brazilian Conference on Intelligent Systems (BRACIS) ([CORRÊA; CARVALHO, 2016](#)).

BIBLIOGRAPHY

ABU-MOSTAFA, Y. S.; MAGDON-ISMAIL, M.; LIN, H.-T. **Learning From Data**. [S.l.]: AMLBook, 2012. ISBN 1600490069, 9781600490064. Cited on page 27.

ALSALLAKH, B.; HANBURY, A.; HAUSER, H.; MIKSCH, S.; RAUBER, A. Visual methods for analyzing probabilistic classification data. **IEEE Transactions on Visualization and Computer Graphics**, v. 20, n. 12, p. 1703–1712, Dec 2014. ISSN 1077-2626. Cited 3 times on pages 49, 56, and 60.

ANKERST, M.; BERCHTOLD, S.; KEIM, D. A. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In: **Proceedings of the 1998 IEEE Symposium on Information Visualization**. Washington, DC, USA: IEEE Computer Society, 1998. (INFOVIS '98), p. 52–. ISBN 0-8186-9093-3. Disponível em: <<http://dl.acm.org/citation.cfm?id=647341.721216>>. Cited on page 40.

AODHA, O. M.; STATHOPOULOS, V.; BROSTOW, G. J.; TERRY, M.; GIROLAMI, M.; JONES, K. E. Putting the Scientist in the Loop - Accelerating Scientific Progress with Interactive Machine Learning. **Proceedings - International Conference on Pattern Recognition**, p. 9–17, 2014. ISSN 10514651. Cited on page 45.

ATTENBERG, J.; IPEIROTIS, P.; PROVOST, F. Beat the Machine: Challenging Workers to Find the Unknown Unknowns. **Human Computation**, v. 1, n. 1, p. 2–7, 2011. ISSN 1936-1955. Disponível em: <<http://www.aaai.org/ocs/index.php/WS/AAAIW11/paper/viewPDFInterstitial/3954/4261>>. Cited on page 68.

BABAE, M.; TSOUKALAS, S.; RIGOLL, G.; DATCU, M. Visualization-Based Active Learning for the Annotation of SAR Images. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, v. 8, n. 10, p. 4687–4698, 2015. ISSN 21511535. Cited 2 times on pages 46 and 56.

BERTINI, E.; DELL'AQUILA, L.; SANTUCCI, G. Springview: cooperation of radviz and parallel coordinates for view optimization and clutter reduction. In: **Coordinated and Multiple Views in Exploratory Visualization (CMV'05)**. [S.l.: s.n.], 2005. p. 22–29. Cited 2 times on pages 61 and 74.

CARO, L. D.; FRIAS-MARTINEZ, V.; FRIAS-MARTINEZ, E. Analyzing the role of dimension arrangement for data visualization in radviz. In: ZAKI, M. J.; YU, J. X.; RAVINDRAN, B.; PUDI, V. (Ed.). **Advances in Knowledge Discovery and Data Mining**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 125–132. ISBN 978-3-642-13672-6. Cited on page 74.

CHAN, W. W.-y. A Survey on Multivariate Data Visualization. **Science And Technology**, n. June, p. 1–29, 2006. Cited 5 times on pages 35, 38, 39, 41, and 74.

CHEN, H.; CHEN, W.; MEI, H.; LIU, Z.; ZHOU, K.; CHEN, W.; GU, W.; MA, K. L. Visual abstraction and exploration of multi-class scatterplots. **IEEE Transactions on Visualization and Computer Graphics**, v. 20, n. 12, p. 1683–1692, 2014. ISSN 10772626. Cited on page 36.

CORRÊA, I. B.; CARVALHO, A. C. P. L. F. d. Dualradviz: Preserving context between classification evaluation and data exploration with radviz. In: **2016 5th Brazilian Conference on Intelligent Systems (BRACIS)**. [S.l.: s.n.], 2016. p. 241–246. Cited 2 times on pages 58 and 74.

DOLEISCH, H. **Visual Analysis of Complex Simulation Data using Multiple Heterogenous Views**. Tese (Doutorado) — Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, out. 2004. Disponível em: <<https://www.cg.tuwien.ac.at/research/publications/2004/doleisch-thesis/>>. Cited on page 61.

DRUMMOND, C.; HOLTE, R. C. Cost curves: An improved method for visualizing classifier performance. **Machine Learning**, v. 65, n. 1, p. 95–130, 2006. ISSN 08856125. Cited 2 times on pages 47 and 56.

GARG, S.; NAM, J. E.; RAMAKRISHNAN, I. V.; MUELLER, K. Model-driven visual analytics. In: **Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on**. [S.l.: s.n.], 2008. p. 19–26. Cited on page 56.

GONCALVES, M. L.; NETTO, M. L. A.; COSTA, J. A. F.; JUNIOR, J. Z. An unsupervised method of classifying remotely sensed images using kohonen self-organizing maps and agglomerative hierarchical clustering methods. **Int. J. Remote Sens.**, Taylor & Francis, Inc., Bristol, PA, USA, v. 29, n. 11, p. 3171–3207, jun. 2008. ISSN 0143-1161. Disponível em: <<http://dx.doi.org.ez67.periodicos.capes.gov.br/10.1080/01431160701442146>>. Cited 2 times on pages 50 and 56.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The elements of statistical learning: data mining, inference and prediction**. 2. ed. Springer, 2009. Disponível em: <<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>>. Cited on page 54.

HEALEY, C. G.; BOOTH, K. S.; ENNS, J. T. Visualizing real-time multivariate data using preattentive processing. **ACM Trans. Model. Comput. Simul.**, ACM, New York, NY, USA, v. 5, n. 3, p. 190–221, jul. 1995. ISSN 1049-3301. Disponível em: <<http://doi.acm.org/10.1145/217853.217855>>. Cited on page 22.

HÖFERLIN, B.; NETZEL, R.; HÖFERLIN, M.; WEISKOPF, D.; HEIDEMANN, G. Interactive learning of ad-hoc classifiers for video visual analytics. In: **Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on**. [S.l.: s.n.], 2012. p. 23–32. Cited 2 times on pages 46 and 56.

HOFFMAN, P.; GRINSTEIN, G.; MARX, K.; GROSSE, I.; STANLEY, E. Dna visual and analytic data mining. In: **Visualization '97., Proceedings**. [S.l.: s.n.], 1997. p. 437–441. Cited on page 40.

IM, J. F.; MCGUFFIN, M. J.; LEUNG, R. Gplom: The generalized plot matrix for visualizing multidimensional multivariate data. **IEEE Transactions on Visualization and Computer Graphics**, v. 19, n. 12, p. 2606–2614, Dec 2013. ISSN 1077-2626. Cited on page 44.

KEIM, D.; KOHLHAMMER, J.; ELLIS, G.; MANSMANN, F. **Mastering the Information Age Solving Problems with Visual Analytics**. [s.n.], 2010. 57 – 86 p. ISSN 1877-0509. ISBN 978-3-905673-77-7. Disponível em: <<http://diglib.org>>. Cited 3 times on pages 21, 22, and 51.

KOHONEN, T. The self-organizing map. **Proceedings of the IEEE**, v. 78, n. 9, p. 1464–1480, Sep 1990. ISSN 0018-9219. Cited on page 50.

KRAUSE, J.; PERER, A.; BERTINI, E. Infuse: Interactive feature selection for predictive modeling of high dimensional data. **IEEE Transactions on Visualization and Computer Graphics**, v. 20, n. 12, p. 1614–1623, Dec 2014. ISSN 1077-2626. Cited 2 times on pages 44 and 56.

KRAUSE, J.; PERER, A.; NG, K. Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models. **ACM Conference on Human Factors in Computing Systems**, 2016. Cited 4 times on pages 24, 54, 55, and 56.

MA, K. L. Machine learning to boost the next generation of visualization technology. **IEEE Computer Graphics and Applications**, v. 27, n. 5, p. 6–9, 2007. ISSN 02721716. Cited 2 times on pages 51 and 56.

MAATEN, L. Learning a parametric embedding by preserving local structure. In: DYK, D. V.; WELLING, M. (Ed.). **Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS-09)**. Journal of Machine Learning Research - Proceedings Track, 2009. v. 5, p. 384–391. Disponível em: <<http://jmlr.csail.mit.edu/proceedings/papers/v5/maaten09a/maaten09a.pdf>>. Cited on page 50.

MAATEN, L. van der. Accelerating t-sne using tree-based algorithms. **Journal of Machine Learning Research**, v. 15, p. 3221–3245, 2014. Disponível em: <<http://jmlr.org/papers/v15/vandermaaten14a.html>>. Cited 2 times on pages 44 and 56.

MAATEN, L. van der; HINTON, G. E. Visualizing high-dimensional data using t-sne. **Journal of Machine Learning Research**, v. 9, p. 2579–2605, 2008. Cited on page 39.

MAHENDRAN, A.; VEDALDI, A. Visualizing deep convolutional neural networks using natural pre-images. **Int. J. Comput. Vision**, Kluwer Academic Publishers, Hingham, MA, USA, v. 120, n. 3, p. 233–255, dez. 2016. ISSN 0920-5691. Disponível em: <<http://dx.doi.org/10.1007/s11263-016-0911-8>>. Cited on page 52.

MARTENS, D.; HUYSMANS, J.; SETIONO, R.; VANTHIENEN, J.; BAESENS, B. Rule extraction from support vector machines: An overview of issues and application in credit scoring. In: _____. **Rule Extraction from Support Vector Machines**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 33–63. ISBN 978-3-540-75390-2. Disponível em: <http://dx.doi.org/10.1007/978-3-540-75390-2_2>. Cited on page 53.

MAYORGA, A.; GLEICHER, M. Splatplots: Overcoming overdraw in scatter plots. **IEEE Transactions on Visualization and Computer Graphics**, v. 19, n. 9, p. 1526–1538, Sept 2013. ISSN 1077-2626. Cited 2 times on pages 36 and 38.

MIGUT, M. A.; WORRING, M.; VEENMAN, C. J. Visualizing multi-dimensional decision boundaries in 2d. **Data Mining and Knowledge Discovery**, v. 29, n. 1, p. 273–295, 2015. ISSN 1573-756X. Disponível em: <<http://dx.doi.org/10.1007/s10618-013-0342-x>>. Cited 3 times on pages 52, 53, and 56.

ONO, J. H. P.; SIKANSI, F.; CORRÊA, D. C.; PAULOVICH, F. V.; PAIVA, A.; NONATO, L. G. Concentric radviz: Visual exploration of multi-task classification. In: **2015 28th SIBGRAPI Conference on Graphics, Patterns and Images**. [S.l.: s.n.], 2015. p. 165–172. ISSN 1530-1834. Cited 3 times on pages 40, 49, and 56.

PACE, K.; BARRY, R. Sparse spatial autoregressions. **Statistics and Probability Letters**, v. 33, n. 3, p. 291–297, 1997. Cited on page 54.

PAIVA, J. G. S.; SCHWARTZ, W. R.; PEDRINI, H.; MINGHIM, R. An approach to supporting incremental visual data classification. **IEEE Transactions on Visualization and Computer Graphics**, v. 21, n. 1, p. 4–17, Jan 2015. ISSN 1077-2626. Cited 3 times on pages 48, 51, and 56.

RADFORD, A.; METZ, L.; CHINTALA, S. Unsupervised representation learning with deep convolutional generative adversarial networks. **International Conference on Learning Representations**, abs/1511.06434, 2015. Disponível em: <<http://arxiv.org/abs/1511.06434>>. Cited on page 52.

RIPLEY, B. D. **Pattern Recognition And Neural Networks**. [S.l.: s.n.], 2008. v. 11. ISBN 978-0521717700. Cited on page 66.

ROSSI, F. Visual Data Mining and Machine Learning. **ESANN'2006 proceedings - European Symposium on Artificial Neural Networks**, n. April, p. 26–28, 2006. Cited 2 times on pages 44 and 50.

ROWEIS, S. T.; SAUL, L. K. Nonlinear dimensionality reduction by locally linear embedding. **Science**, American Association for the Advancement of Science, v. 290, n. 5500, p. 2323–2326, 2000. ISSN 0036-8075. Disponível em: <<http://science.sciencemag.org/content/290/5500/2323>>. Cited on page 42.

RUBIO-SÁNCHEZ, M.; RAYA, L.; DÍAZ, F.; SANCHEZ, A. A comparative study between radviz and star coordinates. **IEEE Transactions on Visualization and Computer Graphics**, v. 22, n. 1, p. 619–628, Jan 2016. ISSN 1077-2626. Cited 2 times on pages 40 and 42.

SAMEK, W.; BINDER, A.; MONTAVON, G.; LAPUSCHKIN, S.; MÜLLER, K. R. Evaluating the visualization of what a deep neural network has learned. **IEEE Transactions on Neural Networks and Learning Systems**, PP, n. 99, p. 1–14, 2016. ISSN 2162-237X. Cited on page 52.

SAMMON, J. W. A nonlinear mapping for data structure analysis. **IEEE Transactions on Computers**, C-18, n. 5, p. 401–409, May 1969. ISSN 0018-9340. Cited on page 42.

SEIFERT, C.; GRANITZER, M. User-based active learning. **Proceedings - IEEE International Conference on Data Mining, ICDM**, p. 418–425, 2010. ISSN 15504786. Cited 2 times on pages 46 and 56.

SELBST, A. D.; POWLES, J. Meaningful information and the right to explanation. **International Data Privacy Law**, v. 7, n. 4, p. 233–242, 2017. Disponível em: <<http://dx.doi.org/10.1093/idpl/ix022>>. Cited on page 23.

SUN, G. D.; WU, Y. C.; LIANG, R. H.; LIU, S. X. A survey of visual analytics techniques and applications: State-of-the-art research and future challenges. **Journal of Computer Science and Technology**, v. 28, n. 5, p. 852–867, 2013. ISSN 10009000. Cited on page 43.

TALBOT, J.; LEE, B.; KAPOOR, A.; TAN, D. S. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**, p. 1283–1292, 2009. Disponível em: <<http://portal.acm.org/citation.cfm?id=1518895>>. Cited 4 times on pages 45, 47, 48, and 56.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining, (First Edition)**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321321367. Cited on page 31.

TENENBAUM, J. B.; SILVA, V. d.; LANGFORD, J. C. A global geometric framework for nonlinear dimensionality reduction. **Science**, American Association for the Advancement of Science, v. 290, n. 5500, p. 2319–2323, 2000. ISSN 0036-8075. Disponível em: <<http://science.sciencemag.org/content/290/5500/2319>>. Cited on page 41.

THERNEAU, T. M.; ATKINSON, E. An introduction to recursive partitioning using the rpart routines. v. 61, 01 1997. Cited on page 66.

VELLIDO, A.; MARTÍN, J. D.; ROSSI, F.; LISBOA, P. J. G. **Seeing is believing: The importance of visualization in real-world machine learning applications**. 2011. Cited on page 38.

VESANTO, J. Som-based data visualization methods. **Intell. Data Anal.**, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 3, n. 2, p. 111–126, mar. 1999. ISSN 1088-467X. Disponível em: <[http://dx.doi.org.ez67.periodicos.capes.gov.br/10.1016/S1088-467X\(99\)00013-X](http://dx.doi.org.ez67.periodicos.capes.gov.br/10.1016/S1088-467X(99)00013-X)>. Cited on page 50.

WACHTER, S.; MITTELSTADT, B. D.; RUSSELL, C. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. **Harvard Journal of Law and Technology**, abs/1711.00399, p. 841–887, 2017. Disponível em: <<http://arxiv.org/abs/1711.00399>>. Cited on page 23.

WARD, M.; GRINSTEIN, G.; KEIM, D. A. **Interactive Data Visualization, Foundation, Techniques, and Applications**. [S.l.: s.n.], 2010. 1–5 p. ISBN 9780874216561. Cited 2 times on pages 36 and 42.

WELLER, A. Challenges for transparency. **ICML Workshop on Human Interpretability in Machine Learning**, abs/1708.01870, 2017. Disponível em: <<http://arxiv.org/abs/1708.01870>>. Cited on page 23.

WITTEN, I. H.; FRANK, E.; HALL, M. a. **Data Mining: Practical Machine Learning Tools and Techniques (Google eBook)**. [s.n.], 2011. 664 p. ISSN 14337851. ISBN 0080890369. Disponível em: <<http://books.google.com/books?id=bDtLM8CODsQC{&}pgi>>. Cited 4 times on pages 29, 31, 33, and 47.

YOSINSKI, J.; CLUNE, J.; NGUYEN, A. M.; FUCHS, T.; LIPSON, H. Understanding neural networks through deep visualization. **Deep Learning Workshop, 31 st International Conference on Machine Learning**, abs/1506.06579, 2015. Disponível em: <<http://arxiv.org/abs/1506.06579>>. Cited 3 times on pages 52, 53, and 56.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: FLEET, D.; PAJDLA, T.; SCHIELE, B.; TUYTELAARS, T. (Ed.). **Computer Vision – ECCV 2014**. Cham: Springer International Publishing, 2014. p. 818–833. ISBN 978-3-319-10590-1. Cited on page 52.

ZHOU, J.; CHEN, F. Making machine learning usable. **International Journal of Intelligent Systems Technologies and Applications**, v. 14, n. 2, p. 91–109, 2015. Cited on page 27.

ZHU, X. Cross-Domain Semi-Supervised Learning Using Feature Formulation. **IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society**, v. 41, n. 6, p. 1627–1638, 2011. ISSN 1941-0492. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/21712162>>. Cited on page 28.