A Comparative Study for the Selection of Machine Learning Algorithms based on Descriptive Parameters

Chettan Kumar, Martin Käppel, Nicolai Schützenmeier, Philipp Eisenhuth and Stefan Jablonski
Institute for Computer Science, University of Bayreuth, Universitätsstraße 30, Bayreuth, Germany

Keywords: Machine Learning, Algorithm Recommendation, Data Analysis, Information System.

Abstract: In this paper, we present a new cheat sheet based approach to select an adequate machine learning algorithm. However, we extend existing cheat sheet approaches at two ends. We incorporate two different perspectives towards the machine learning problem while simultaneously increasing the number of parameters decisively. For each family of machine learning algorithms (e.g. regression, classification, clustering, and association learning) we identify individual parameters that describe the machine learning problem accurately. We arrange those parameters in a table and assess known machine learning algorithms in such a table. Our cheat sheet is implemented as a web application based on the information of the presented tables.

1 INTRODUCTION

The development of algorithms for machine learning tasks has made large progress over the last years. Nowadays, there is a very large number of different algorithms for data analysis problems. Consequently, often the hardest part of solving a machine learning problem is finding the right algorithm for the job. Obviously, it would be ideal to have vast knowledge about strengths and weaknesses of all these algorithms. However, this is almost not possible due to the incredible number of different algorithms. Also, it is good to know from experience when to use which algorithm. Alternatively, a general approach for finding the best fitting algorithm is to run the most known ones and select the three to four best performing ones for further testing. Obviously, this is a very time consuming and tedious work without a deeper understanding of the underlying causes.

To overcome the problem of selecting an adequate machine learning algorithm, several cheat sheets are provided on the internet. The idea is to ask a domain expert some questions about the machine learning task and then recommend one or more algorithms. For ease of use these cheat sheets are kept very simple. That means they only focus on a small set of parameters (mostly size of data and labelling) and ignore important properties of algorithms like the risk for overfitting or the handling of different types of noise. With regard to the complexity of machine learning problems we consider such cheat sheets as not adequate to cope with a decent selection of a machine learning algorithms (see Section 3). In addition, there are many interesting research papers (Kotsiantis, 2007; Xu and Tian, 2015), which compare different machine learning algorithms and work out strengths, weaknesses and parameter based recommendations with respect to specific machine learning tasks. Nevertheless, the information about these machine learning algorithms is spread over many papers and this is very difficult to grasp.

In this paper, we present a new cheat sheet based approach to select an adequate machine learning algorithm. Therefore, we extend existing cheat sheet approaches at two ends. First, we incorporate two different perspectives towards the machine-learning problem while simultaneously increasing the number of parameters decisively. While the first perspective provides an application-oriented view, the second perspective focusses on a set of technical parameters about the underlying data, algorithms, and their usage. The application-oriented view contributes experience of solving similar problems by describing the problem with the relevant parameters and the applied algorithm. The increased number of parameters describing either the application or the technical perspective enables a more realistic and accurate selection of a machine learning algorithms since both problem and solutions are described in much more detail. In the best case scenario, the application perspective and the technical perspective of the cheat sheet search and propose algorithms, respectively, and the

Kumar, C., Käppel, M., Schützenmeier, N., Eisenhuth, P. and Jablonski, S.
A Comparative Study for the Selection of Machine Learning Algorithms based on Descriptive Parameters.
DOI: 10.5220/0008117404080415
In *Proceedings of the 8th International Conference on Data Science, Technology and Applications (DATA 2019)*, pages 408-415
ISBN: 978-989-758-377-3

intersection of both proposals is not empty. The algorithms lying in the intersection should then depict the best or a set of best fitting algorithms for the problem to solve. Otherwise, we get a list of algorithms that does not need to be further tested. With respect to knowledge, experience, and available time, it is also possible to ignore one of the perspectives at the expense of accuracy.

Our second extension of conventional cheat sheets is the overall structure of our cheat sheet. For each family of machine learning algorithms (e.g. regression, classification, clustering, and association learning) we identify individual parameters that describe the machine learning problem accurately. We arrange those parameters in a table and assess known machine learning algorithms in such a table. Since we expect that further parameters, which might be very helpful in assessing the adequacy of a specific algorithms, can be added easily as the tables are extensible. Typically, the labelling of these algorithms in that table is based on an extensive literature review and on practical experiences made in numerous data analysis projects. It is worth to mention that it is also possible to simplify the cheat sheet with respect to the experience of the user or the available information about the data set by leaving out some columns (parameters). In contrast to the hierarchical structure of conventional cheat sheets (see Section 3) not all parameters have to be evaluated for a recommendation.

The rest of this paper is structured as follows. Section 2 gives an overview of related work. In Section 3 we analyse three different existing cheat sheets. Afterwards, in Section 4, we define relevant parameters for the analysis of machine learning algorithms. In Section 5 and Section 6, we present our cheat sheet and give reasons for the judgment of the labelling of the used parameters. Sections 7 and 8 conclude the main results of the paper and give an overview of further work.

2 RELATED WORK

Selecting an adequate algorithm for a machine learning problem is a difficult task due to the vast amount of available algorithms. Hence, working out strengths and weaknesses is an essential part of a large number of research papers. In (Kotsiantis, 2007) several classification techniques are evaluated with regard to accuracy, speed, tolerance to missing values, redundant attributes, dealing with danger of overfitting or noise. As a rating system the authors use a scale from one to four points, describing best (four points) and worst performance (one point). In (Arora et al., 2013) and (Kumbhare and Chobe, 2014) some of the more common algorithms for association learning (like Apriori, AIS, SETM and so on) are evaluated and an overview of strengths and weaknesses are given. In (Ruiand D. Wunsch, 2005) Xu gives a detailed survey about clustering algorithms and their application. A theoretical background is given and the complexity and the capability of tackling high dimensional data for the algorithms is evaluated. In (Xu and Tian, 2015) the survey is expanded by comparing their capability for data size, the shape of model and their sensitivity to noise and outliers. Note that in many research papers no distinction is made between noise and outliers. In (Duwe and Kim, 2017) supervised learning algorithms are tested on different data sets and their accuracy and speed is measured by AUC, Precision, Recall and ACC.

3 ANALYSIS OF EXISTING CHEAT SHEETS

This section gives an overview on some existing graphical cheat sheets that should help the user choosing an adequate algorithm for a given machine learning problem. We criticize the following deficiencies in all these cheat sheets: First, the number of attributes that contribute to the recommendation is not very high. Second, the properties of the algorithms under selection, like the handling of missing values and noise or the risk of overfitting, are completely ignored. Besides, the authors made no distinction between the data-oriented and application-oriented perspective and association learning is not covered at all. They all suffers from the problem that if a specific question cannot be answered the user gets stuck and cannot progress any further. This happens due to the mostly hierarchical structure. Hence, the cheat-sheets often cannot give a good recommendation. In the following part, we discuss cheat sheets from Microsoft Azure, Scikit-learn, and SAS in depth to discuss the above-mentioned deficiencies in detail.The cheat sheet from Microsoft-Azure[1] covers four main categories of algorithms: regression, classification, clustering, and anomaly detection. It has a hierarchical structure where the paths leading to the categories are dependent on the goal of the application. A concrete algorithm within a category is then selected by only one additional decision, which oftentimes is not very meaningful either, because it only looks at properties like the accuracy and training time of the algo-

[1] https://docs.microsoft.com/de-de/azure/machine-learning/studio/algorithm-cheat-sheet

Table 1: Classification table ([1]=(Ke et al., 2017), [2]=(Chen and Guestrin, 2016), [3]=(Dorogush et al., 2017), [4]=(Freund and Schapire, 1997), [5]=(Hastie et al., 2009), [6]=(Kotsiantis, 2007), [7]=(Bramer, 2013), [8]=(Aggarwal, 2014)).

Algorithm	**TOD**	**DOD**	**FN**	**IN**	**RN**	**HO**	**SOTD**
Logistic Regression	num				Y	N	L
	[8]				[8]	[8]	[8]
Naive Bayes	cat		N	Y	Y	Y	
	[7]		[6]	[6]	[6]	[6]	
Artificial Neural Network	num		N	N	N	N	
	[6]		[6]	[6]	[6]	[6]	
k-Nearest-Neighbour	cat			N	N		S
	[7]			[6]	[6]		[6]
Support Vector Machine	num		Y	N		N	
	[6]		[6]	[6]		[6]	
Decision Tree	cat, num			Y	Y		S
	[6]			[6]	[6]		[6]
Random Forest	cat, num	L	Y		Y	Y	L
	[5]	[5]	[5]		[5]	[5]	[5]
AdaBoost	cat	L	N	N	N		L
	[4]	[4]	[4]	[4]	[4]		[4]
CatBoost	cat	L		Y	Y	L	
	[3]	[3]		[3]	[3]	[3]	
XGBoost	num	L			Y	Y	L
	[2]	[2]			[2]	[2]	[2]
LightGBM	cat	L			Y	Y	L
	[1]	[1]			[1]	[1]	[1]

rithms. Additionally, distinct aspects of the problems are covered by the various decisions, so it is hard to compare the algorithms among each other. The available data is almost never part of the decision. The only considered aspect is the number of features. All in all, this cheat sheet is not profound enough and ambiguous. Scikit-learn also offers a graphical cheat sheet[2]. It covers four main categories of algorithms: regression, classification, clustering and dimensionality reduction. Here, the resulting recommendation is based upon a chain of binary decisions that mainly considers the size and type of the dataset. The cheat sheet has some dead ends with no alternatives and only covers a small number of different algorithms. The last cheat sheet under consideration is offered by SAS[3]. Here, the algorithms are first grouped by the learning paradigm (supervised and unsupervised). Then they are further categorized by clustering and dimension reduction for unsupervised learning and classification together with regression for supervised learning. Like with the former discussed two cheatsheets, the number of decisions leading to the final algorithms is quite small and the quality of the branches varies in regard of the possible choices and the quality of the answers.

[2]https://scikit-learn.org/stable/tutorial/machine_learning_map

[3]https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/

4 PARAMETERS

In this section, we describe different parameters that are useful for the recommendation of appropriate algorithms and we want to use in our cheat sheet. On the one hand we have parameters which are important for every algorithm, e.g. the dimension of the data set. On the other hand we have certain approach-based parameters which are only significant for some algorithms, e.g. the shape of a model is only relevant for clustering algorithms. In the following we define and explain all the parameters that are used throughout the paper.

Type of Data (TOD): This parameter indicates if the algorithm handles numerical (num), categorical (cat) or spatial data. Most algorithms do not work with mixed types. Hence a preprocessing to convert the data into the required type is necessary.

Dimension of Data Set (DOD): Indicates if the chosen algorithm can work with a large (L) or a small (S) number of features in the data set.

Outliers (O): An outlier is a record in the data set which has at least one value that is very high or very low compared to the values of all the other data points. We associate an algorithm with Y if it works well with a data set with more than 20% outliers.

Table 2: Clustering table ([1] = (Xu and Tian, 2015), [2]=(Tiruveedhula et al., 2016), [3]=(Ruiand D. Wunsch, 2005).

Algorithm	**SOD**	**DOD**	**O**	**TOD**	**SOM**	**Based**
k-Means	L [1]	S [1],[3]	N [1]	num [2]	convex [1]	Partition [1]
PAM	S [1]	S [1]	Y [1]	cat, num [2]	convex [1]	Partition [1]
CLARA	L [1]	S [1],[3]	Y [1]	num [2]	convex [1]	Partition [1]
CLARANS	L [1]	S [1],[3]	Y [1]	num [2]	convex [1]	Partition [1]
BIRCH	L [1]	S [1],[3]	Y [1]	num [2]	convex [1]	Hierarchical [1]
CURE	L [1]	L [1],[3]	Y [1]	num, cat [2]	arbitrary [1]	Hierarchical [1]
ROCK	L [1]	L [1]	Y [1]	cat [2]	arbitrary [1]	Hierarchical [1]
Chameleon	L [1]	L [1]	Y [1]	num, cat [2]	arbitrary [1]	Hierarchical [1]
DBSCAN	L [1]	S [1],[3]	Y [1]	num [2]	arbitrary [1]	Density [1]
OPTICS	L [1]	S [1]	Y [1]	num [2]	arbitrary [1]	Density [1]
DENCLUE	L [1]	L [1][3]	Y [1]	num [2]	arbitrary [1]	Density [1]
STING	L [1]	S [1]	Y [1]	spatial [2]	arbitrary [1]	Grid [1]
Wave	L [1]	S [3] [1]	Y [2]	num [2]	arbitrary [2]	Grid [1]
CLIQUE	L [1]	S [3] [1]	Y [1]	num [2]	arbitrary [1]	Grid [1]
SOM	S [1]	L [1]	Y [1]	num, cat [2]	convex [1]	Model [1]
SLINK	L [2]	S [2]	N [2]	num [2]	arbitrary [2]	Model [1]

Otherwise we denote it with N.

Feature Noise (FN): Indicates if the chosen algorithm takes measures to work with features that do not help explaining the target. In other words: irrelevant or weak features.

Item Noise (IN): Indicates if the chosen algorithm works well with anomalies, like wrong or missing values, in certain data items.

Record Noise (RN): Indicates if the chosen algorithm works well with records, which do not follow the form or relation, which most of the records do.

Handling of Overfitting (HO): Indicates if the chosen algorithm takes measures to avoid overfitting.

Size of Training Data (SOTD): Indicates if the chosen algorithm needs a large (L) or a small (S) training data set to generate a well-performing model. As aforementioned this judgment also depends on the use case.

Size of Data Set (SOD): Indicates if the chosen algorithm works well with large (L) or small (S) data sets. If a data set is described as large or small always depends on the use case, e.g. 10000 image records would be large whereas the same number of sensor measurements would be small.

Multicollinearity (MC): Indicates if the chosen algorithm works well with multicollinearity. Multicollinearity is a phenomenon in which two or more predictor variables in a regression model are highly correlated, meaning that one can be linearly predicted from the others with a substantial degree of accuracy.

Based: Describes the basic concepts behind the clustering. We distingiush between model-, density-, grid- and hierarchical-based clustering algorithms.

Shape of Model (SOM): Describes the form of the resulting clusters, e.g. convex or rectangle clusters.

5 RELEVANT MACHINE LEARNING TECHNIQUES

In this section we present our main result, a table based cheat sheet for machine learning algorithms. Therefore, we payed attention on four different techniques: Classification, Clustering, Regression and Association Learning. The following chapters describe these four techniques and present the tables with are essential for our developed cheat sheet. An empty entry means that it is either hard or even impossible to fill the cell or we did not find a reliable source to justify our decision.

5.1 Classification

Classification is an approach of supervised learning technique for modeling and predicting categorical variables(Bramer, 2013). The classification includes two phases. The first phase is a learning process phase in which the training data is analyzed, then the rules and patterns are created. The second phase is the real classification. The in phase 1 trained model is used to predict the class for the new, not known data points. Our cheat sheet approach for the Classification is shown in table 1.

5.2 Clustering

Clustering is an unsupervised learning technique that involves the grouping of data points that are in some properties identical to one another (Bramer, 2013). It is a common technique for statistical data analysis that used in many different fields. Basically, clustering algorithms are used to classify each point in the data set into a specific group based on the properties or features. In data analysis, clustering is used to gain some useful information when we do not have labeled data like supervised learning and we are aiming at finding out, in which group data points fall when we apply a clustering algorithm. Our cheat sheet approach for the Classification is shown in table 2.

5.3 Regression

Regression analysis is an important approach for modeling and analyzing the relationship between a dependent and one or more independent variables (often also called explanatory variable). The case of one independent variable is called simple linear regression, meanwhile for more than one independent variable the approach is called multiple linear regression. In both cases the resulting model is called linear model, because the dependent variable is expressed as a linear combination of the regression coefficients. It is used for forecasting, time series modeling and finding the causal effect relationship between the variables. Our cheat sheet approach for the Regression is shown in table 3.

5.4 Association Learning

The main usage of association rule learning is for knowledge discovery in transactional databases. There (mostly) categorical data of grouped items (so called item sets) is analyzed and association rules are derived. These rules consider the most associated items from the dataset based on predefined threshold values for a support and confidence parameter. The support parameter for a specific item set reflects the relative amount of item sets in which it occurs compared to the whole dataset. However, the confidence parameter describes the relative amount of item sets in which the right-hand-side of a specific rule occurs given that the left-hand-side of the rule is already fulfilled (Agrawal et al., 1993). Association rules can be helpful in many different application fields like basket data analysis, cross-marketing, catalog design, web usage mining, intrusion detection, continuous production and bioinformatics. In basket data analysis for example, a supermarket collects information about the shopping behavior of its customers. By looking at products which are purchased together frequently, valuable information can be obtained. This can then be used to increase the revenue for the market with the support of a good marketing strategy based on the discovered knowledge. The AIS (Agrawal et al., 1993) and SETM (Houtsma and Swami, 1995) algorithms were early developed algorithms for mining such association rules from a given dataset. With the introduction of the Apriori algorithm (together with AprioriTid and the refined version AprioriHybrid, which uses the best parts from both the normal Apriori and AprioriTid), the performance of finding association rules was significantly improved by the clever usage of properties for frequent item set generation (Agrawal and Srikant, 1994). Eclat (Zaki, 2000) (short for “equivalence class transformation”) further reduced the time needed for deriving the rules mainly by the usage of a vertical database schema for calculating support and confidence values and therefore reducing the needed database scans. The FP-growth algorithm avoids the costly candidate generation and uses a special FP-tree (frequent pattern tree) instead for deriving the association rules. This improves the runtime even further, because only two database scans are needed for the tree generation (Han et al., 2000).
Since the presented algorithms are always an im-

Table 3: Regression table([1]= (Chen and Guestrin, 2016), [2]=(Tibshirani, 1996), [3]=(Montgomery et al., 2015), [4]=(Breiman, 2001), [5]=(Hastie, 2017), [6]=(Breiman, 1996a), [7]=(Rodriguez and Yao, 2013), [8]=(Breiman, 1996b), [9]=(Unger et al., 2009), [10]=(Elith et al., 2008), [11]=(Huang et al., 2006), [12]=(Stone, 1985), [13]=(Friedman, 1991), [14]=(Drucker et al., 1997), [15]=(Griggs, 2013), [16]=(Marsh and Cormier, 2011), [17]=(Winship and Mare, 1984), [18]=(Murphy, 2012), [19]=(Hinde, 1982), [20]=(Elster et al., 2015), [21]=(N. van Wieringen, 2015), [22]=(Wulu et al., 2002), [23]=(Zou and Hastie, 2005), [24]=(Koenker and Bassett, 1978), [25]=(Koenker and Hallock, 2001), [26]=(Luo et al., 2015)).

Algorithm	**SOD**	**DOD**	**O**	**HO**	**MC**
Linear Regression	S [7]	S [3]	N [7]	N [3],[18]	N N [3]
Quantile	L [24],[25]	S [24],[25]	Y [24]		N [3]
Bayesian	L [20]	S [18],[20]	Y [18]	Y [20],[18]	
LASSO	L [2]		Y [2]	Y [2]	Y [2]
Ridge	L [18]	L [18],[21]	Y [3],[18]	Y [3],[18]	Y [3],[21]
Elastic Net	L [18],[23]	L [18],[23]	Y [18]	Y [23]	Y [18],[23]
Ordinal	L [17]	L [17]	Y [17]	Y [17]	
Poisson	L [19],[22]	L [19],[22]		Y [19],[22]	
SVR	S [14]	L [14]	Y [14]	Y [14]	
Spline	L [15]			Y [16],[15]	Y [16],[15]
MARS	L [13]	L [15],[13]		Y [13]	Y [15],[13]
Additive	L [12]	L [12]		Y [12]	
RF	L [4]	L [4]	Y [4]	Y [4]	
Extreme Learning	L [11]		Y [11]	N [26]	Y [11]
Ensemble	L [9],[8]		Y [9],[8]	Y [9],[8]	Y [9]
Boosted	L [6],[10]		Y [6],[10]	Y [6],[10]	
XGBoost	L [1]	L [1]	Y [1]	Y [1]	

provement of the previously presented algorithms, it is hard to make a fair and useful comparison. The application of association rule mining is strongly determined by the concrete use case. Therefore, the data already has to be in the right format to apply the association learning algorithms at all. The quality of the results from all these algorithms is the same (for same support and confident thresholds), which is why the runtime (and maybe the memory usage) is the only property for comparison. The most promising algorithms considering the runtime needed for finding association rules within a specific dataset are FP-growth and Eclat, because they are the most refined algorithms. All the other algorithms are now mainly useful for educational purposes. There are still some algorithms which were not considered here, since they are not very well known and they provide no improvement performance-wise.

6 IMPLEMENTATION

Our implementation of the cheat sheet is based on the tables of the previous section. The content of the tables is stored in a database together with additional information like the sources, comments and further recommendations. We implemented our cheat sheet as a web application[4], which has access to the database for algorithm recommendation. The implementation

[4]Demo will be available at: https://www.ai4.uni-bayreuth.de/en/research/ToolsAndResources/

is divided into four different parts. The main part of the cheat sheet is the algorithm recommendation tool, where an adequate algorithm is proposed based on the problem description and some properties of the data set. Here the information which is entered by the user regarding the properties gets transformed into a corresponding query to the underlying database. For later maintenance and further extensions, we have implemented a graphical user interface that makes it easier to edit algorithms and their properties. The application dialog enables every user to contribute his experience into the cheat sheet. All deposited information can be viewed in the comparison section of the application. Algorithms can be listed or sorted by different criteria and can be compared to each other to inform oneself or for getting a manual algorithm recommendation.

7 FUTURE WORK

Despite of the large number of evaluated algorithms and parameters, some table entries still remain blank and need to be evaluated in a future work. Moreover, some fields of machine learning, like deep learning or reinforcement learning were not evaluated yet. The evaluation of different algorithms in this paper is based on an exhaustive literature review, but an experimental validation is still missing. Beyond the specific task of algorithm comparison, there are many other open problems that will need to be tackled if we want to make true progress in solving the grand task of algorithm recommendation: The judgement of the parameters depend on so many different factors (e.g. context, the strength of the parameter or the preprocessing), that often an evaluation with ”yes” or ”no” is not sufficient or misleading. Hence we should try to work out conditions for the evaluation and include them into our cheat sheet. Therefore, the comment possibility in our web application should be improved. A further promising research area for algorithm recommendation is the analysis of successfully completed machine learning problems. Based on a concise description of the problem and the solving strategy we want to transfer this knowledge to similar machine learning problems. The cheat sheet should be extended by increasing the number of analysed projects. For consistent evaluation of the data set properties, an automatic detection of the important parameters would also be useful.

8 CONCLUSION

In this paper we have extended existing cheat sheets at two ends to a new table based cheat sheet. We increased the number of parameters in comparison to the existing cheat sheets decisively and added a knowledge base to include human experience from other similar machine learning problems. We evaluated the most common algorithms of classification, clustering, regression and association learning by an exhaustive literature review. Therefore, we choose for each type of machine learning important properties to evaluate strengths and weaknesses for the algorithms. A simple judgement with “yes” or “no” respectively “large” or “small” simplifies the algorithm recommendation because the user must only decide between two different options. For the ease of use and better expandability and maintenance, we built a web application. This application provides a simple query tool for algorithm recommendation and various input fields for updating parameters, adding algorithms or already solved machine learning problems.

REFERENCES

Aggarwal, C. C. (2014). *Data Classification: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition.

Agrawal, R., Imielinski, T., and Swami, A. (1993). *Mining Association Rules Between Sets of Items in Large Databases, SIGMOD Conference*, volume 22, pages 207–.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB ’94, pages 487–499, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Arora, J. B., Bhalla, N., and Rao, S. J. (2013). A review on association rule mining algorithms.

Bramer, M. (2013). *Principles of Data Mining*. Springer Publishing Company, Incorporated, 2nd edition.

Breiman, L. (1996a). Bias, variance , and arcing classifiers. *Technical Report 460, Statistics Department, University of California.*

Breiman, L. (1996b). Stacked regressions. *Machine Learning*, 24(1):49–64.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, pages 785–794, New York, NY, USA. ACM.

Dorogush, A. V., Ershov, V., and Gulin, A. (2017). Catboost: gradient boosting with categorical features support. *CoRR*, abs/1810.11363.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., and Vapnik, V. (1997). Support vector regression machines. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*, pages 155–161. MIT Press.

Duwe, G. and Kim, K. (2017). Out with the old and in with the new? an empirical comparison of supervised learning algorithms to predict recidivism. *Criminal Justice Policy Review*, 28(6):570–600.

Elith, J., Leathwick, J. R., and Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecology*, 77(4):802–813.

Elster, C., Klauenberg, K., Walzel, M., Wübbeler, G., Harris, P., Cox, M., Matthews, C., Smith, I., Wright, L., Allard, A., Fischer, N., Cowen, S., Ellison, S., Wilson, P., Pennecchi, F., Kok, G., van der Veen, A., and Pendrill, L. (2015). A guide to bayesian inference for regression problems.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *Ann. Statist.*, 19(1):1–67.

Griggs, W. (2013). Penalized spline regression and its applications.

Han, J., Pei, J., and Yin, Y. (2000). Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 1–12, New York, NY, USA. ACM.

Hastie, T. (2017). Generalized additive models. In Fagerberg, J., Mowery, D. C., and Nelson, R. R., editors, *Statistical models in S*, chapter 7, pages 249–307. Routledge.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition.

Hinde, J. (1982). Compound poisson regression models. In Gilchrist, R., editor, *GLIM 82: Proceedings of the International Conference on Generalised Linear Models*, pages 109–121, New York, NY. Springer New York.

Houtsma, M. and Swami, A. (1995). Set-oriented mining for association rules in relational databases. In *Proceedings of the Eleventh International Conference on Data Engineering*, pages 25–33.

Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1):489 – 501. Neural Networks.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc.

Koenker, R. and Hallock, K. (2001). Quantile regression. *Journal of Economic Perspectives*, 15(4):143–156.

Koenker, R. W. and Bassett, G. (1978). Regression quantiles. *Econometrica*, 46(1):33–50.

Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24. IOS Press.

Kumbhare, T. A. and Chobe, S. V. (2014). An overview of association rule mining algorithms.

Luo, X., Chang, X., and Ban, X. (2015). Extreme learning machine for regression and classification using l1-norm and l2-norm. In Cao, J., Mao, K., Cambria, E., Man, Z., and Toh, K.-A., editors, *Proceedings of ELM-2014 Volume 1*, pages 293–300, Cham. Springer International Publishing.

Marsh, L. and Cormier, D. R. (2011). Spline regression models. *Journal of Applied Business Research (JABR)*, 19.

Montgomery, D. C., Peck, E. A., and Vining, G. G. (2015). *Introduction to Linear Regression Analysis*. John Wiley & Sons, New York.

Murphy, K. P. (2012). *Machine Learning - A Probabilistic Perspective*. MIT Press, Cambridge.

N. van Wieringen, W. (2015). Lecture notes on ridge regression.

Rodriguez, R. N. and Yao, Y. (2013). Five things you should know about quantile regression. In *In Proceedings of the SAS Global Forum 2017 Conference*.

Ruiand D. Wunsch, X. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678.

Stone, C. J. (1985). Additive regression and other nonparametric models. *Ann. Statist.*, 13(2):689–705.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288.

Tiruveedhula, S., Sheela Rani, C., and Narayana, V. (2016). A survey on clustering techniques for big data mining. *Indian Journal of Science and Technology*, 9:1–12.

Unger, D. A., van den Dool, H., O'Lenic, E., and Collins, D. (2009). Ensemble regression. *Monthly Weather Review*, 137(7):2365–2379.

Winship, C. and Mare, R. D. (1984). Regression models with ordinal variables. *American Sociological Review*.

Wulu, J., Singh, K., Famoye, F., Thomas, T., and McGwin, G. (2002). Regression analysis of count data. *Journal of Indian Society of Agricultural Statistics*, 55:220–231.

Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193.

Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):372–390.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320.