USC University of Southern California

Decision Tree and Random Forest Algorithms: Accuracy, Complexity & Scalability In Breast Cancer Tumor Classification

Course: INF 552 - Machine Learning
Section: Tues, 3:30-6:50pm
Contributors: Mark Mann - mbmann@usc.edu; Bridget Haus - bhaus@usc.edu; Peter Argo - pargo@usc.edu

Abstract

Decision tree classification applied to breast cancer testing shows strong promise as a tool for aiding pathologists in diagnosis. As the prevalence of breast cancer remains high, healthcare providers will look to decrease medical costs through early and accurate diagnosis of the disease. To enable more accurate test results, an opportunity exists for decision trees to aid pathologists in the diagnosis process. Decision tree models allow for explainable results, which can lead to further trust and adoption by the wider pathology community. However, implementing decision tree models into current diagnosis systems and software raises some questions. How will these models perform when training data becomes large and is constantly changing? In our paper, we will evaluate decision tree and random forest methodology, accuracy, and time complexity using the UCI Breast Cancer Dataset. We compare the results of Gini vs. Entropy in determining the best split. In Random Forests, we evaluate parameter options against time and accuracy in attempts of finding the best performing version given a big data environment. Furthermore, we compare performance metrics of Support Vector Machine (SVM) and Neural Network models to understand model trade-offs.

Breast Cancer Dataset

The UCI Breast Cancer Dataset [6] represents 30 computed features from digitized images of 569 fine needle aspirate (FNA) tests. Each training image contains a set of cells from the biopsy and a label of ‘benign’ or ‘malignant’. In this dataset, it is assumed that the relevant cell tissue was recovered during biopsy procedure by a trained physician. Another assumption is that each training image was properly framed by a domain expert (pathologist). These assumptions must be fulfilled to represent the biopsy data properly. Each derived feature describes the shape of the cells in the image (Figure 1). Examples of the derived features include: radius_mean, perimeter_mean, and compactness_mean. It is worth noting this dataset does not consider demographic nor genetic features of the patient (i.e. age, race, BRCA1 status). Adding these additional features from a patient’s electronic medical record (EMR) would enable further hypothesis testing. We use decision tree modeling to evaluate the data and compare results with similar machine learning analysis produced by support vector and neural network algorithms.

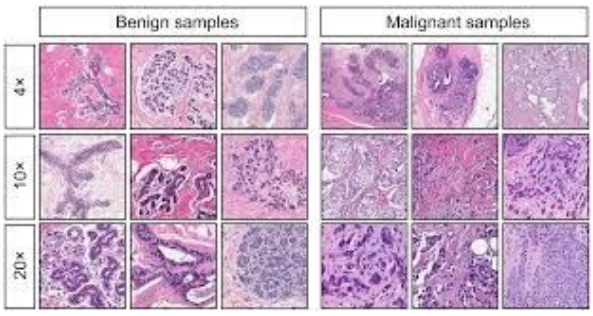


Figure 1: Benign and Malignant images

Decision Tree Mechanics

Decision trees are a popular and widely used supervised learning technique for classification and regression tasks. The basis of decision tree algorithms are built upon the major concept of class purity, where the algorithm ultimately seeks to divide the data such that asking questions can correctly predict the label outcome. [1] A good node will ask a question that takes sets of heterogeneous labels and splits them into subsets of homogeneous labels, thus classifying the data. In order to construct the nodes of the tree, both entropy and gini index are two popular techniques for measuring class purity.

In information theory, entropy describes the disorder and uncertainty of a system. [2] Entropy has an inverse relationship with probability, such that if $P(X) = 1$ then $I(X) = 0.$ This is because if the probability is 100%, there is no uncertainty in the random variable and therefore no entropy. Entropy measures the expected amount of information gained by knowing the outcome of a random trial. [2] For our research problem, we were interested in a specific subset of this postulation known as the binary entropy function. The Bernoulli process is modeled as a random variable X that can only take the values true or false. In our breast cancer dataset, our labels indicate that the tumor is either malignant or benign, so we can model the entropy in our challenge problem as $I(X) = -plog_2(p) - (1-p)log_2(1-p)$ where our random variable X is the tumor and $P(X = malignant) = p$ and $P(X = benign) = 1 - p.$

Gini impurity takes a different approach by measuring the likelihood of an incorrect classification of a new instance given that instance was randomly assigned according to the distribution of labels from the dataset. [8] This can be calculated as $I_G(n) = 1 - \sum_{i=1}^{C}(p_i)^2$ where p_i is the probability of classification i. In this method, when the gini measurement is zero we have achieved perfect classification because the summed probability of misclassification is zero.

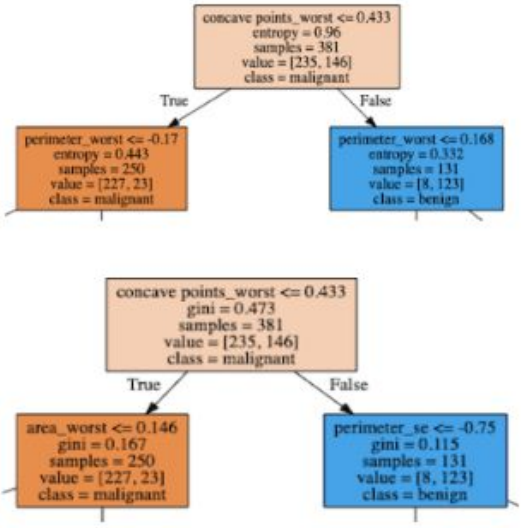


Figure 2: Root nodes split on entropy (above) and gini(below)

Regardless of the method chosen, ultimately the decision tree learning algorithm examines all attributes, decides which has the best information gain, and then selects that attribute as the root node. [2] Overall, the best decision tree algorithm depends on the given dataset. Gini split favors larger partitions, while entropy favors partitions that have small counts but many distinct values. [8] We applied a python scikit library to our dataset, and generated two decision trees using each criterion. They both produced a tree with a root node of concave points_worst <= 0.433, where concave points are the number of indentations present on the nuclear border of the tumor. [3] Figure 2 displays the root nodes of our decision trees, one with an entropy calculation of 0.96 and one with a gini measurement of 0.473. Appendix A & B contain the full decision trees for splitting on entropy and gini respectively.

Overfitting is when a model learns the training data too well, and is thus not generalized well enough to adequately perform on new test instances. When a model is overfit, it captures the fluctuation in noise of the training data set and uses that noise as learned concepts in the final model. [15] Overfitting is more likely in nonparametric models, where the number of parameters is able to grow with the sample size. [9] In decision trees, a small sample of tumors may only require a small tree, but a large sample of tumors may require a very large tree to achieve the same accuracy. Because the size of the tree is variable rather than fixed, this is a nonparametric model. [9] One way to avoid overfitting a tree is by setting a max tree depth. This specifies that although a pure class has not been reached in our training set, we would like to arrive at our prediction anyway. This is a method used to avoid creating a large and complex tree that represents the training set perfectly so that it may have a more generalized shape when it comes to unseen test instances. As we can see in the figure to the left, the decision tree achieves its maximum accuracy with a maximum depth of six, and then does not improve further for each subsequent layer. This means we can drastically simplify the tree without sacrificing accuracy.

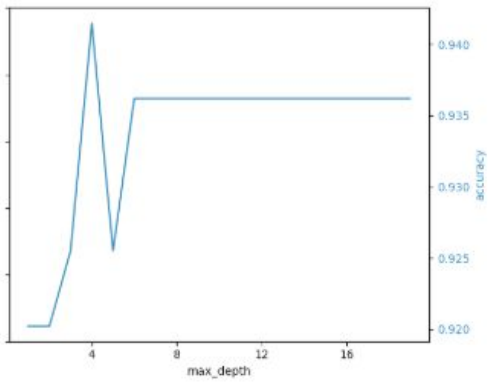
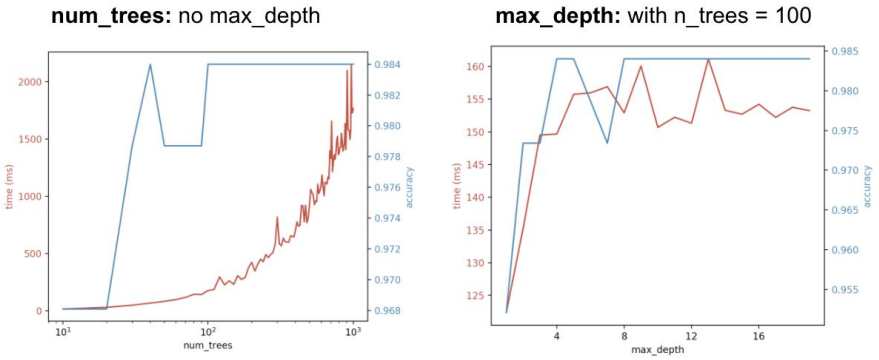


Figure 3: Model Accuracy with respect to maximum depth size of the decision tree

An interesting characteristic of the binary decision tree learning algorithm is that its method of calculation is an NP-complete problem. In theoretical computer science, P are problems that can be solved efficiently, where efficiency is defined in polynomial solving time related to the input size. [4] NP are all problems in which, if a solution is given, one can verify or prove that the solution is correct in polynomial time. NP-complete are known as a class of problems that are the hardest within NP as they can be reduced to problems of similar difficulty. [4] Specifically, binary decision trees are shown to be in the NP-complete class through a reduction from the Exact Cover 3 problem which is proven in a 1976 paper titled, *Constructing Optimal Binary Decision Trees Is NP-Complete.* [14] It is important to know the algorithmic complexity because as real medical professionals seek to incorporate machine learning into their discipline, there needs to be assurances that the algorithm is scalable.

Random Forest

An extension of decision tree modelling is the random forest ensemble method. Random forests are a means to reduce overfitting from a single decision tree by building multiple decision trees from bootstrapped versions of the same dataset. In bootstrapping, new datasets (called ‘bags’) are generated at random and allow replacement from the training set. From each bag, a decision tree is trained and can be further randomized to only consider a few random features at each node split. [10] After all trees are grown, the ensemble of trees are used together to make a classification. The label with the highest amount of ‘votes’ from the ensemble of trees is ultimately selected. This means for any testing sample, a classification is made in O(m*max_depth) time, where m is the amount of trees (Figure X). The random forest model is flexible, as more trees can be added to the pre-existing model as more training data becomes available.



To obtain a random forest with low training time and high accuracy, cross-validation is conducted. Two parameters to consider are a) number of trees and b) max depth. To identify the best set of parameters, training time and accuracy are visualized against these parameters (Figure 4).

Figure 4: Random forest performance with varying num_trees and max_depth

For the breast cancer dataset, the optimal number of trees and maximum depth is 100 and 10 respectively. However, as the available training data grows, the optimal parameters values may change. When training data grows and changes rapidly, cross-validation checks must occur frequently to ensure the optimal model parameter values are selected. Random forest models are well-poised for updates to training data. For example, scripts can be written to drop poor-performing trees and new trees can be added instead of re-growing the whole forest. Finally, new decision trees can be grown in parallel which significantly reduces the time complexity of updating the model. [11]

Other Classifiers: Support Vector Machine and Neural Network

Support vector machine algorithms are a means of classifying instances of a dataset by determining a marginal classifier that optimally separates each instance based on its classification. The shortest distance between the support vector classifications and the separation hyperplane is called the margin. In order to avoid high variance, soft marginal classifiers that allow for some misclassifications, and classifications that are within the margin, are identified through cross validation. If the data is not linearly separable, a kernel function is applied that would produce a necessary separation of the data set to determine the support vector classifier[16]. Support vector machines (SVMs) are a commonly used methodology for image interpretation and medical classification tasks among other things.

Neural network algorithms are inspired by the functionality of the human brain such that they absorb input training data to determine patterns within said data and ultimately predict the outputs for a new instance of the dataset. Neural networks utilize layers of neurons as the core processing units of the network. Each instance of data is fed to each neuron in the first layer. Neurons of one layer are connected to neurons of the next layer via channels. Each channel is initially arbitrarily assigned a weight that is multiplied by the output of the first layer. Each neuron is associated with a bias that is added to the input. With this, an activation function is applied to the layer to determine which neurons are “activated”, which determines the neurons that transmit data to the next layer. This methodology is continued through each layer in a method called forward propagation. Once at the final layer, the network will determine which classification the instance belongs to based on highest probability. Using

the training data, the predicted classification is compared with the actual classification. Typically high error rates occur early in the training of the network due to an arbitrary initial assignment of weights and biases. The error is then transferred back through the network in a method called back propagation, and the weights are adjusted accordingly [17]. Artificial neural networks are also currently utilized for streamlining medical diagnostics to help avoid misdiagnosis.

Model Comparison Results

For each classification model, the training time and accuracies are considered as baseline metrics for comparison. In considering training time, model time complexity formulas are considered below.

Training and Run Time Complexities of Classifiers

	Training time	Run time	Model terms
Decision Tree	O(n*log(n) * d)	O(max depth)	n - rows d - features m - trees s - support vectors e - epochs i - input nodes j - hidden layer nodes k - output nodes
Random Forest	O(m* n*log(n) * d)	O(m*max depth)	
SVM - Linear	$O(n^2)$	O(s*d)	
NN	O(en(ij + jk))	O(i)	

Results

	Parameters	Training Time (ms)	Accuracy
Decision Tree (scikit_tree.py)	train_size = 66% max_depth=None	4.34	0.936
Random Forest (random_forest.py)	train_size = 66% max_depth=10 num_trees=100	150.69	0.984
SVM - Linear (scikit_svm.py)	train_size = 66% kernel=linear	2.26	0.984
NN (scikit_nn.py)	train_size = 66% epochs=1000 hiden_nodes =100 activation=logistic	495.80	0.98

Figure 5: Time complexities and performance analysis of four machine learning models

For our relatively small dataset (569 rows), the SVM model produced the highest accuracy with the lowest training time. However, the SVM time complexity increases in quadratic time $O(n^2)$, which becomes prohibitively large as the training size increases. Within a large testing ecosystem (commercial labs, health system testing facility, etc.), the size of training images will grow large, so only models that perform well with large input data must be considered.

The next best model was random forest, which achieved the same accuracy with slightly higher training time. This training time can be reduced significantly with additional machines, as new trees in the random forest can be grown in parallel [11]. As data changes, there is more flexibility to update random forest models. Updates to random forests simply involve adding new trees or dropping pre-existing trees. However, updates to SVMs are typically limited to linear SVMs with current support from only a few packages [12]. Neural networks can be trained by increasing the number of epochs, which is defined as one cycle through the full training dataset. However simple feed-forward networks are prone to the ‘catastrophic interference’ problem. In this problem, newly learned data interferes with old learned data causing tests on old data to perform worse [13]. To solve this, more complex ‘Long Short Term Memory’ (LSTM) models can be employed. However, the LSTM still must be trained on one machine and does not scale as efficiently as random forest for our particular dataset.

Conclusion

Each of the evaluated machine learning algorithms provides a highly effective means of classifying breast cancer images. While the SVM algorithm produced highest accuracy, the random forest and neural network maintained performance results that are comparable and within an uncertain margin that is dependent on the input test set. We used accuracy of classifying instances in our test set as our

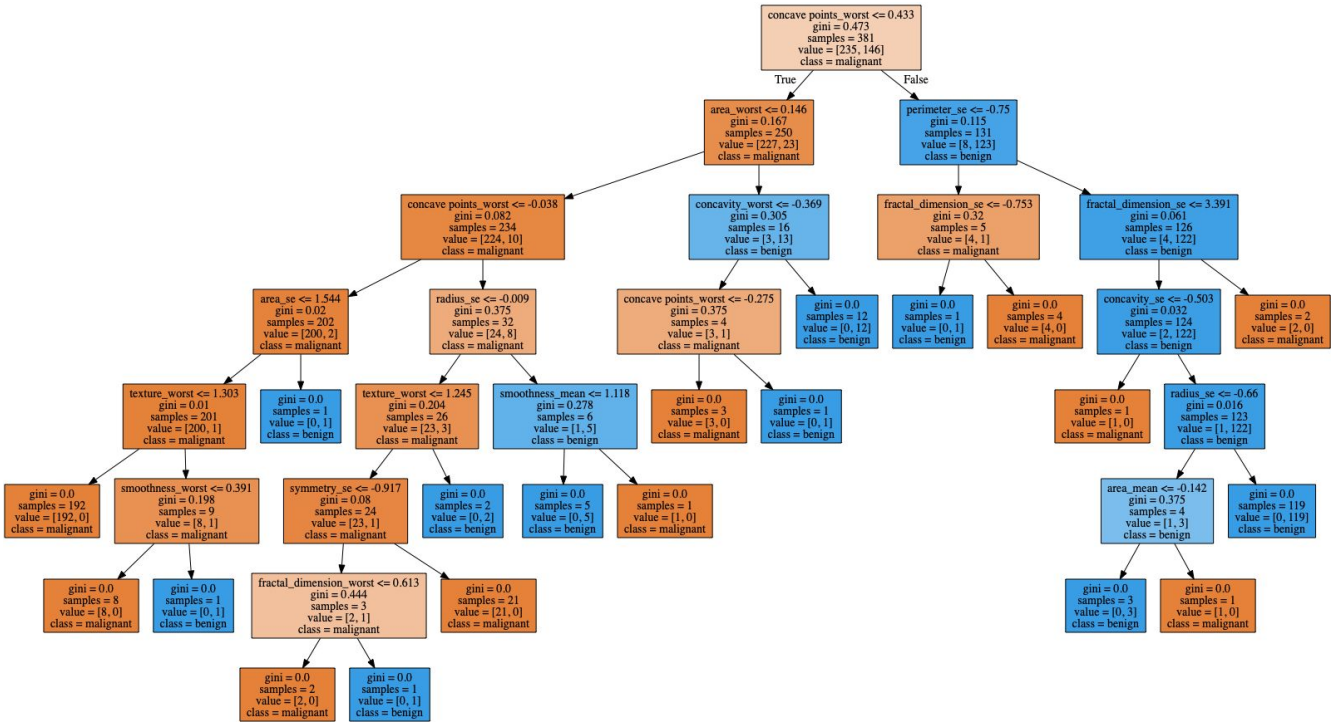
first method of performance analysis. Accuracy is determined by considering the total number of true positives and true negatives and dividing that sum by the total number of instances in the test set.

Each algorithm maintains respective advantages and disadvantages that should be considered when analyzing a dataset. Neural networks, and all deep learning algorithms, are widely considered the most powerful method of performing abstractions and feature extractions for image processing tasks. However, neural networks require a relatively sizable training dataset to develop an effective network of neurons and thus requires a considerable amount of time and processing capability to model the training data. Moreover, it is difficult to determine the optimal architecture of the network and therefore mandates a greater expertise to debug and analyze. Support vector machines yielded optimal results because they are effective in high dimensional spaces. SVM's identify support vectors from a small subset of data points and are therefore memory efficient and computationally friendly. Understanding this, it is reasonable that we derived a training time that is orders of magnitude less than that of the random forest and neural network. A disadvantage to SVM's, however, is that it does not provide a direct probability and requires an expensive k-fold cross validation to determine this.

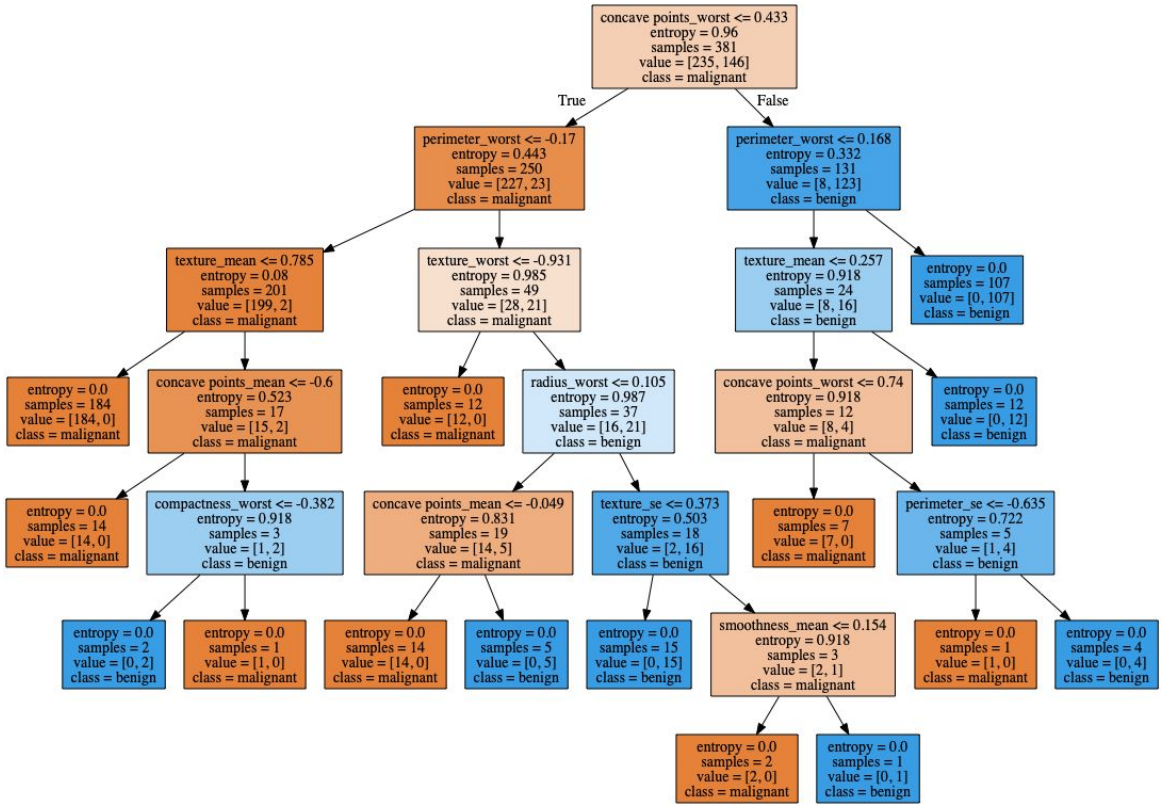
Some advantages of random forests include a minimal need for preprocessing, the ability to handle missing values, an impressive efficacy on high-dimensional data, and low sensitivity to hyperparameter tuning. However, random forest trees are sub-optimal for regression analysis, require higher memory allocation for saving each tree and surrogate-split, and demand high computation capability to process each datapoint through each tree. As previously stated, the dataset maintained high dimensionality (30 computed features) and was limited in size (569 images). It is thus reasonable that random forests performed nearly optimally in making predictions for the test dataset.

Moving forward, projects primarily aimed at aiding pathologists in diagnostics medicine should investigate recall in addition to accuracy; that is evaluating correct true positive classifications by the total ture positive classifications. Understanding that the scope of this project was of machine learning nature, we primarily focused on investigating the capability of a decision tree and random forest model. Such an investigation utilized a breast cancer dataset and compared performance results with the classification results of SVM and neural network models. Therefore, we elected to use accuracy and training time as our guiding metrics to most concisely synthesize our findings. However, a more thorough investigation into optimizing recall through tuning parameters / hyperparameters, would be suggested in more medically focused analyses. This analysis would be ideal because medical ethics dictates that a greater emphasis should be placed identifying all instances of malignant images rather than simply producing the most accurate model that has instances of misclassified true malignant images.

Appendix A: Full Decision Tree Entropy Image



Appendix B: Full Decision Tree Gini Image



USC University of Southern California

References

1. "Information theory and decision tree." 2 Mar. 2020, https://cs.nju.edu.cn/wujx/teaching/10_IT.pdf.
2. "Satish Kumar Thittamaranahalli - USC - Viterbi School of Engineering." 28 Jan. 2020. *Lecture Notes*.
3. "Supplementary Materials and Methods - Cancer Prevention ...." http://cancerpreventionresearch.aacrjournals.org/highwire/filestream/36568/field_highwire_adjunct_files/1/methfigleg.pdf.
4. "Where the Really Hard Problems Are - IJCAI." https://www.ijcai.org/Proceedings/91-1/Papers/052.pdf.
5. Breast Cancer Statistics. 2020. https://www.cancer.net/cancer-types/breast-cancer/statistics
6. UCI Breast Cancer Dataset. 1994. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
7. Breast Cancer Diagnosis and Prognosis via Linear Programming. 2009. https://dollar.biz.uiowa.edu/~street/research/aim94.pdf
8. Decision Tree Flavors: Gini Index and Information Gain. 2016. http://www.learnbymarketing.com/481/decision-tree-flavors-gini-info-gain/
9. Parametric and Nonparametric Machine Learning Algorithms. 2016. https://machinelearningmastery.com/parametric-and-nonparametric-machine-learning-algorithms/
10. Jake VanderPlas. Python Data Science Handbook. 2017. https://jakevdp.github.io/PythonDataScienceHandbook/05.08-random-forests.html
11. A Parallel Random Forest Algorithm for Big Data in a Spark Cloud Computing Environment. 2018. Cornell. https://arxiv.org/abs/1810.07748
12. Incremental and Decremental SVM Learning. 2000. https://isn.ucsd.edu/pub/papers//nips00_inc.pdf
13. Catastrophic Interference in Connectionist Networks. 1989. https://www.sciencedirect.com/science/article/pii/S0079742108605368
14. Constructing Optimal Binary Decision Trees Is NP-Complete. 1976. https://people.csail.mit.edu/rivest/HyafilRivest-ConstructingOptimalBinaryDecisionTreesIsNPComplete.pdf
15. Overfitting and Underfitting With Machine Learning Algorithms. 2016. https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/
16. “An introduction to Statistical Learning” pg. 337 - 364; An Introduction to Statistical Learning
17. Neural networks and deep learning