MWSUG 2024 - Paper 59

Charting Your Organization's Machine Learning Roadmap

Ryan Paul Lafler, Premier Analytics Consulting, LLC

ABSTRACT

Machine learning is experiencing a golden age of investment, democratization, and accessibility across all domains in the life sciences, natural sciences, and social sciences with applications to industry for business decision-making, risk management, consumer marketing, clinical trials, financial forecasting, security recognition, video remastering, digital twin simulations, and more. But what exactly is machine learning (ML)? How is it connected to Artificial Intelligence (AI)? And most importantly, how can data scientists, programmers, software engineers, and/or researchers start their endeavors into machine learning? This paper answers these questions, and more, by providing a roadmap to help navigate the complexities of machine learning in an application-oriented guide. This paper covers the main aspects of machine learning including supervised, unsupervised, and semi-supervised approaches as well as deep learning. The roadmap for supervised machine learning starts with linear regression and progressively builds towards more complex and flexible algorithms with discussions about the advantages and disadvantages of using certain models over others. This paper discusses the real-world applications of both labeled and unlabeled data; supervised and unsupervised machine learning algorithms; overfitting and underfitting; cross-validation; and the importance of hyperparameter tuning to better fit algorithms to their data.

INTRODUCTION

Machine learning represents a multidisciplinary domain for training, evaluating, automating, extracting, and generating insights from structured and unstructured data sources; labeled and unlabeled datasets; block, file, and object cloud storage systems; or data possessing concrete, flexible, or non-existent schemas. Machine learning can handle a variety of data structures and formats including tabular datasets denoted by their rows and columns to multi-dimensional arrays indexed by location and time. The wide-spread adoption and growth of machine learning reflects the insatiable human curiosity to better understand our respective domains and fields with data-driven, automated approaches.

The goal of this paper is to help readers chart their machine learning roadmaps so that they can navigate the challenging, winding roads of machine learning without careening off the metaphorical cliff.

We'll start with the fundamentals. Before driving off on our journey, let's have a basic roadmap in-hand by answering a few general questions. *What is the difference between machine learning, Artificial Intelligence (AI), and deep learning? How do we define machine learning algorithms and how are they trained? How do algorithms learn from the data?* We'll examine the domains of machine learning and discuss the advantages, limitations, and potential applications that each ML domain can provide to organizations.

With our roadmap in-hand, we'll proceed on a test drive, charting our way through several popular supervised machine learning algorithms and explaining them in terms of model complexity vs. model interpretability. In doing so, we'll avoid speeding off the road because of overfitting and ensure that we're not blocking traffic by underfitting. Safe and proven model evaluation strategies will help you monitor your algorithm's training and prevent reckless driving.

In charting your machine learning roadmap, you and your organization can better understand the ideas, applications, and strategies that are motivating the rapid adoption, research, and development of machine learning techniques to build AI workflows in the 21st century.

1. CHARTING YOUR ROADMAP: UNDERSTANDING ML FUNDAMENTALS

1.1) CONNECTING ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, AND DEEP LEARNING

Artificial intelligence is a broad multidisciplinary field that allows for the innovation, invention, and integration of technologies where systems learn, acquire, and/or mimic human decision-making capabilities. Machine learning represents one potential pathway to achieve Artificial Intelligence, where the machine trains on the provided data, attempts to minimize the number of mistakes with respect to an objective (loss) function, and estimates the best parameters for that specific algorithm.

Deep learning is a specialized form of machine learning where structured, semi-structured, and unstructured data are transformed, engineered, and outputted through a series of connecting layers in robust networks. Neural networks, a common approach to deep learning, are composed from these layers and are implemented for predictive, forecasting, data mining, and generative tasks.

Owing to its democratization from the growth of APIs, libraries, packages, and routines in languages including Python, R, SAS, Java, Julia, C++, Scala, and others, machine learning's wide-spread availability is making it easier for organizations to incorporate into their predictive and decision-making workflows. Examples of these libraries include the Scikit-Learn machine learning library in Python, the cross-language TensorFlow library for deep learning, the Keras library for Python-centric deep learning, and the cross-language XGBoost library.

1.2) MACHINE LEARNING ALGORITHMS, PARAMETERS, AND HYPERPARAMETERS

As we've discussed, machine learning is a method for a system to attain artificial intelligence by using training data to define functions, classification boundaries, generative distributions, clusters, decompositions, relationships, and forecasts that answer research problems without direct human intervention.

Bu how does machine learning work under the hood? The answer lies with programmatic algorithms that best try to achieve these goals:

1. Optimizes an objective (loss) function to minimize some form of error (i.e., MSE, log-loss, Gini Impurity, etc.)
2. Autonomously adjusts internal parameters that control how the algorithm fits to the data (uses the loss function)
3. Defines its own sets of decision rules, and automatically updates these rules, to best achieve the desired task

Together, these items are connected and dependent on each other, comprising the necessary components that allow machine learning algorithms to identify a task, computationally learn from the data, and improve through iterations of model training and refitting. In doing so, algorithms define their own sets of rules that best address the research problem of interest.

Parameters inside of an algorithm are like the *weights* of a linear regression model. Both represent internal components that automatically update based on the training data, and with the given hyperparameters, provide the best possible fit. Parameter values are estimated entirely from the data, conditional on the current set of hyperparameter choices, and can drastically change depending on the training data flowing into the model.

Hyperparameters control the entire configurations of machine learning algorithms. Critically, these meta-parameters control whether the algorithm overfits or underfits; how flexible the model is in unraveling complex relationships; the number of parameters and their values estimated by the data; how quickly the algorithm learns from the data; and more. Hyperparameters are unique to each algorithm and need to be fine-tuned to find the best configuration given your data, objectives, and research question.

2. OFF-RAMPS TOWARDS SUPERVISED AND UNSUPERVISED ML METHODS

2.1) SUPERVISED MACHINE LEARNING

Datasets that contain a set of *target features* are often referred to as *labeled datasets* that are used in training supervised machine learning algorithms to accomplish classification and regression tasks. These algorithms include linear regression models, logistic regression models, naïve bayes generators, decision trees, gradient-boosted trees, random forests, and support vector machines.

These algorithms train from a set of predictors and learn the mapping between them at the target feature of interest. *Supervised algorithms are only as accurate as their set of target features*. Misclassified and/or incorrectly recorded labels can cause supervised algorithms to extract incorrect information, thereby misrepresenting relationships and failing to generalize towards unseen data when deployed for production.

Supervised algorithms fit to their datasets by minimizing some target (loss) function. For a typical linear regression, this loss function is the mean squared error (MSE) which minimizes the squared differences between predicted and actual target observations. In a binary classification problem, where for instance a logistic regression model is trained to classify observations into exactly 2-categories, the classifier iteratively refits the model with parameters until the loss function decreases closer to 0.

Labeled datasets are often more expensive, time-consuming, and labor-intensive to make than their raw data counterparts. Raw datasets, often called *unlabeled datasets*, lack defined target features for supervised algorithms to learn mappings from.

2.2) UNSUPERVISED MACHINE LEARNING

When data lacks any well-defined target feature for a model to learn and map relationships from, this data is called *raw* or *unlabeled data*. This is where unsupervised machine learning algorithms can automate the following tasks:

- Perform data mining to extract abstract features, representations, and insights from the data
- Reduce the dimensionality of data into its principal (latent) space; can address the *curse of dimensionality* issue
- Map associations between similar and dissimilar observations
- Cluster observations; perform anomaly detection; and object detection
- Identify, detect, and separate data signals from random noise
- Natural language processing (NLP) and computer vision (CV) applications

Raw data is everywhere. Organizations tend to have an abundance of unprocessed, raw data that can be leveraged to generate decision-making insights for AI workflows through unsupervised machine learning algorithms. However, these algorithms require external validation checks from actual people. Since there are no labeled records to validate the results from unsupervised algorithms with, external validation must be completed by subject-matter experts and domain specialists to investigate the results and test their quality against what is currently known and the insights these unsupervised algorithms provide.

Popular unsupervised algorithms and applications include K-Means clustering, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), agglomerative clustering, principal components analysis, manifold learning, sentence segmentation for NLP, association rule mining, and more.

3. NAVIGATING THE WINDY ROAD OF SUPERVISED ML ALGORITHMS

3.1) MODEL INTERPRETABILITY VS. MODEL COMPLEXITY

Different algorithms will produce different results. Even when tackling the same classification or regression problem, two different supervised algorithms can produce drastically different results.

Model interpretability represents how easy an algorithm is to understand. Furthermore, the data scientist can understand important characteristics about how the algorithm trained. This includes information on which features are ranked as most important, how decision rules (*splits*) were generated, assumptions restricting the model's degrees of freedom (loosely speaking), and how the results generalize to broader populations of interest.

As models grow more sophisticated, and therefore capable of mapping non-linear, multi-dimensional relationships between the set of predictors and response feature(s), the model becomes less interpretable. Before settling on any machine learning algorithm, data scientists should balance the needs of their algorithm with:

1. Predictive performance
2. Model assumptions (how much freedom the algorithm has when finding predictive solutions)
3. Feature (predictor) importance

A model with fewer assumptions attached to it results in increased predictive performance. Less assumptions means the algorithm has greater freedom to map and learn complex relationships. Models with more assumptions increase interpretability, making relationships between predictors and their target feature(s) easier-to-understand.

3.2) SUPERVISED MACHINE LEARNING ROADMAP

Taking model complexity and interpretability into account, let's propose a suggested roadmap that begins with a hypothesis asking some research question. The destination represents the final, chosen algorithm. When building any predictive model, some good advice is to *start simple and incrementally scale upwards towards more sophisticated, powerful models*. Starting simple and then progressively building towards more advanced algorithms supplies the ML team and organization with baselines to work with—these baselines serve as benchmarks to quantify the potential advancements, limitations, and drawbacks of adopting more powerful (and complex) ML algorithms.

When travelling, the reader may have been reminded to *enjoy the journey*. By training, evaluating, and comparing metrics (i.e., mean squared error, mean absolute error, accuracy, precision, recall, F1 score) against different trained algorithms, an effective comparative analysis can be made to quantify the predictive gains from more "advanced" models in comparison to their "simpler" counterparts. Travelling this journey to select the *best available* algorithm is critical to meeting an organization's needs when predicting and/or interpreting the results from their chosen predictive model.

Figure 1 proposes a *suggested* roadmap that begins with an initial hypothesis (question) that the organization wants to predict, which is then followed by training easy-to-interpret baseline models, incrementally scaling up towards more flexible algorithms, and finally ends when some algorithm performs sufficiently better than the baseline models, without being overly complex, and is selected for deployment.

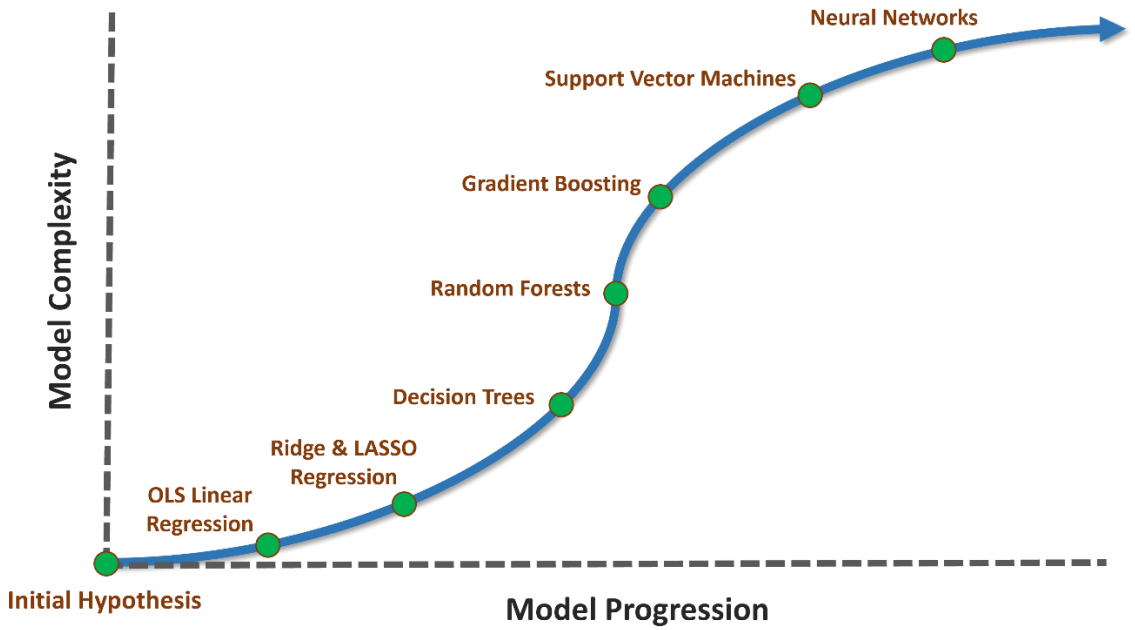

Figure 1. *Supervised machine learning algorithms roadmap (Credit: Ryan Paul Lafler, Premier Analytics Consulting)*

This process begins with ordinary least-squares (OLS) regression. This is a statistical model that utilizes the mean squared error as its loss function and models a numeric response feature against a set of numeric and categorical predictors. OLS regression makes a major assumption about your model's relationship: the predictive function is *linear*. By restricting your model to only detect and estimate the *best-fitting* linear relationship, you are constraining it to be simple, intuitive, and easy-to-interpret. Changes in the response (target) feature can be interpreted by changes in the coefficients (parameters) of your OLS model. The entire model is a weighted summation of features that can include interaction effects and polynomial terms to build more flexible models.

Ridge, LASSO, and elastic-net linear regression algorithms are structurally the same as OLS regression models but include an important difference in their objective (loss) functions. These models possess a *penalization term* that is added to their overall loss functions to mitigate the risk of overfitting by reducing (or eliminating) the effects of miscellaneous predictors in the model through a process referred to as *regularization*. For instance, when a linear model contains several interaction effects and polynomial terms, there exists a significant risk of overfitting to the training data and failing to generalize to unseen data. Regularization terms balance this by introducing *bias* to linear and non-linear models that penalize the inclusion of extraneous terms failing to contribute predictive power to the model. These regularization penalties can be added to any of the algorithms listed in the roadmap proposed in ***Figure 1*** since they only modify the algorithm's loss (objective) function.

Decision trees are versatile, flexible, and easy-to-interpret algorithms that perform classification and regression tasks. Often, decision trees serve as base models to machine learning ensembles like random forests and gradient-boosted models. Random forests help minimize overfitting while gradient-boosted ensembles minimize underfitting by improving on the previous model's errors. Both random forests and gradient-boosted models utilize hundreds, thousands, or even millions of models aggregated together to arrive at some predictive result.

Algorithms that are extremely powerful in detecting non-linear, complex relationships include support vector machines (SVMs) and deep learning architectures like neural networks. SVMs require significant data pre-processing to create classification boundaries or regression functions from small datasets possessing large quantities of features. By employing a technique called the *kernel trick*, SVM's are adept at modeling non-linear, multi-dimensional relationships without having to explicitly transform existing data into higher-dimensional representations (saving computation time and system resources). SVMs are even capable of handling datasets where the number of features greatly exceeds the number of observations.

Neural networks are powerful algorithms capable of modeling complex non-linear relationships using a variety of structured, semi-structured, and unstructured data. Making no prior assumptions about the data, neural networks trade interpretability for superior predictive power, with deeper networks allowing the approximation of complex regression functions, classification boundaries, and distributions. Neural networks are often referred to as 'black boxes', the outdated name given to flight recorders that monitor cockpit conversations and flight data installed in airplanes, due to their hidden layers that transform, modify, and configure the data to the desired output. Trying to interpret the connections between layers' neurons is a very difficult task and are therefore termed 'black boxes' because of their opaque inner workings. Neural networks can approximate non-linear multi-dimensional functions, classification boundaries, and distributions, giving them supreme versatility in AI workflows requiring abstract data transformations, decompositions, generative tasks, and predictive analytics. Neural networks can also handle combinations of structured (i.e., relational database tables), unstructured (i.e., images, videos), and semi-structured (i.e., flat files) data sources within the same model, making them versatile in leveraging both labeled and unlabeled data simultaneously.

4. WATCHING YOUR SPEEDOMETER: TOOLS FOR EVALUATING TRAINING

4.1) MODEL EVALUATION THROUGH CROSS-VALIDATION

Algorithms evaluated on the same data they were trained on will always result in biased results favoring its performance. Why does this happen? Because after training on the same dataset, estimating the parameters that best fit that dataset, and learning insights from that dataset, model evaluation metrics using the training data will always paint a more pristine picture than what the algorithm truly understands.

A similar analogy for this phenomenon is a student memorizing material from an exam preparation study guide and learning useless information (noise) such as the order of multiple-choice questions before the actual exam. The exam represents *unseen data*, with the student's memorization equating to the ML algorithm's training process. Once the student memorizes the study guide's content and noise (i.e., the order of questions, number of multiple-choice questions), without understanding how to apply that content beyond their study guide, that student then fails their exam and receives a terrible final score. The student, much like an ML algorithm learning useless noise inside of its training data, suffers from over-reliance on the study guide and as a result, overfits to the study guide and fails to extrapolate their learned insights towards the actual exam.

Machine learning algorithms need to understand their training data. Furthermore, these algorithms need to extrapolate their learned insights beyond the training data to unseen data they have not yet encountered. This is referred to as *algorithm generalization* and requires unbiased methods to achieve it.

One such method includes randomly shuffling the original data (assuming the data is not dependent on time or spatial location) and partitioning it into a mutually exclusive training and testing set. This training-testing split typically reserves 80% of the shuffled data for training, leaving the remaining 20% of the shuffled data in a *holdout* testing set that the algorithm doesn't encounter until the final evaluation stage.

This method is simple and easy-to-implement, but still fails to provide an unbiased picture of an algorithm's performance across different segments of the data. Cross-validation, on the other hand, partitions the randomly shuffled dataset into a pre-determined K-number of folds, with K being a positive integer selected by the machine learning engineer, where different folds of the data are used to train the algorithm. This results in K-number of trained algorithms being created from the partitioned dataset.

For example, ***Figure 2*** shows a cross-validation design containing 5-folds, where the data is partitioned into a series of training and testing folds that generates a total of 5-distinct models. Notice how every fold of data is used as a testing set when training the 5-distinct models. The blue-shaded folds comprise that model's training dataset, while the green-shaded folds encompass all the observations set aside to form that model's testing (evaluation) set.

By implementing techniques such as 5-folds cross-validation, we're able to train several models across different combinations of observations from the same dataset and evaluate these models' predictive performances (using their respective test sets) to quantify the variability in predictive metrics and calculate an overall average prediction score from the results of each model.

	Fold #1	Fold #2	Fold #3	Fold #4	Fold #5
Model #1					Test Set
Model #2				Test Set	
Model #3			Test Set		
Model #4		Test Set			
Model #5	Test Set				

Figure 2. *5-Folds cross-validation for model evaluation (Credit: Ryan Paul Lafler, Premier Analytics Consulting)*

CONCLUSION

This paper serves as a self-start guide to help anyone interested in adopting, integrating, and/or experimenting with ML algorithms to develop AI workflows capable of automation and data-driven decision-making. Machine learning is a powerful means of achieving data-driven AI systems that can automate data mining, insight extraction, predictive modeling, and generative processes across multidisciplinary fields and industries.

From defining the fundamental components of machine learning algorithms, to discussing supervised and unsupervised approaches, understanding the model complexity—interpretability tradeoff, and examining techniques to better evaluate model performance during and after training, this paper develops a machine learning roadmap that organizations, professionals, students, educators, and enthusiasts alike can all benefit from when developing data-driven AI workflows.

REFERENCES

Brown, S. (2021, April 21). *Machine learning, explained*. MIT Sloan. https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

Géron, A. (2022). *Hands-on machine learning with scikit-learn, keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems* (3rd ed.). O'Reilly.

Lafler, R. P., & Wade, A. (2024, May 19—22). Unraveling the layers within neural networks: Designing artificial and convolutional neural networks for classification and regression using python's keras & tensorflow. *Real World Evidence and Big Data[Lecture]*. PharmaSUG 2024, Baltimore, Maryland. https://www.lexjansen.com/pharmasug/2024/RW/PharmaSUG-2024-RW-390.pdf

ACKNOWLEDGEMENTS

The author would like to extend his appreciation to the Midwest SAS Users Group (MWSUG) 2024 Conference, the Conference Committee, the Academic Chair and the Operations Chair, and the Section Chair for accepting this paper. The author expresses his gratitude to the MWSUG 2024 Conference for giving him the opportunity to present, publish, and network with professional colleagues, industry specialists, researchers, and students at this esteemed data science conference.

ABOUT THE AUTHOR

Ryan Paul Lafler is the Founder, CEO, Chief Data Scientist, and Lead Consultant at **Premier Analytics Consulting, LLC**, a data science consulting firm based in San Diego, California. He's also an Adjunct Faculty Member and Research Scientist at San Diego State University (SDSU) for the Big Data Analytics Graduate Program, the Department of Mathematics and Statistics, and the Climate Informatics Laboratory. Ryan's programming experience in Python, R, SAS, JavaScript (React.js), open-source API frameworks, and SQL has contributed to his success as a big data scientist; consultant; ML engineer; statistician; and full-stack application developer. He received his Master of Science in Big Data Analytics from San Diego State University in May 2023 following the publication of his thesis. He holds a Bachelor of Science in Statistics, a minor in Quantitative Economics, and graduated *magna cum laude* from San Diego State University. Ryan's passionate about applied machine learning, deep learning, Artificial Intelligence, statistics, full-stack application and interactive dashboard development, data visualization, and open-source programming languages.

CONTACT INFORMATION

Comments, suggestions, and/or any questions are encouraged and may be sent to:

Ryan Paul Lafler, M.Sc.
Premier Analytics Consulting, LLC *and* San Diego State University
Founder, CEO, Chief Data Scientist, Lead Consultant, and Adjunct Faculty
E-mail: rplafler@premier-analytics.com
Website: www.Premier-Analytics.com
LinkedIn: www.LinkedIn.com/in/RyanPaulLafler
Résumé: www.Premier-Analytics.com/ryan-paul-lafler