Paper SAS108-2023

Introduction to Machine Learning: Descriptions and Best Practices

Jim Box, SAS Institute

ABSTRACT

Artificial Intelligence and Machine Learning (AI & ML) are THE hot topics these days. In this talk, we'll define ML and go over the many types of models and explain how they work and what they are used for. We'll also cover the best practices around using ML models, including how to identify the best model. Finally, we will explore some of the concerns about implementing ML models - specifically covering how to analyze the model process for human biases that can have dramatic impacts on society.

INTRODUCTION

Discussions about Artificial Intelligence (AI) and Machine Learning (ML) are everywhere. You've probably heard your company leaders say things like they need to use AI/ML to improve processes. Maybe you saw that Microsoft is making a big investment in OpenAI, the company behind the ChatGPT application. AI & ML are terms that are used interchangeably, but do have different meanings:

Artificial Intelligence is the development of systems that can perform tasks which typically require human intelligence, such as speech recognition, visual perception, as decision making, all done without human intervention. AI systems have been in place since the mid 1950s, when the RAND corporation created something called the Logic Theorist, which could prove mathematical theorems using symbolic logic. AI systems don't always have complicated mathematics under the hood, often they rely on well-defined rules to make decisions. One of the earliest applications of AI in medicine was the MYCIN project from Stanford University in the 1970s, which was used to identify sources of bacterial infections and recommend antibiotic regimen. It walked the physician through several yes/no questions about the patient and provided a ranked list of likely bacterial culprits.

Machine Learning refers to algorithms that train computer systems to improve their performance on specific tasks by learning from data, without explicitly being programmed with rules on how to do so. ML algorithms are able to learn from experience, make predictions and decisions, and improve their performance over time. ML is often considered a branch of AI. ML algorithms have also been in operation since the mid 1950s, with one of the early algorithms being the Perceptron, which was used to classify input data into two categories. This early form of a neural network (more on those later) was developed in 1958.

At this point, you may be wondering what the difference between ML and statistical modeling is. While there is a fair amount of overlap in the two fields, but there are some key differences (Table 1).

	Statistical Modeling	Machine Learning
Data	Data has some underlying probability distribution	Wide range of data types
Goals	Test hypotheses / explore relationships	Make predictions / decisions
Approach	Specify a model and estimate parameters	Learn patterns by itself
Scope	Smaller data with a specific question	Large, complex data with many variables
Model Validation	Check the underlying assumptions about the data	See how the model does on data from a hold-out sample

Table 1. Statistical Modeling Approach vs Machine Learning Approach

Machine Learning also has slightly different nomenclature from what you may recall from your statistics class (Table 2).

Statisticians Say …	ML Practitioners Say …
Variable	Feature
Independent Variable / Predictor	Input
Dependent Variable	Target
Observation	Object

Table 2. Differences in Nomenclature

Machine Learning algorithms are classified as different types of learning. The types you typically hear the most about are Supervised Learning, Unsupervised Learning, Deep Learning and Reinforcement Learning.

SUPERVISED LEARNING

Supervised learning is a type of ML algorithm that is trained on a labeled dataset. That means the model is given data where the value of the outcome/prediction is known and figures out on its own how to predict the target variable based on the values of the inputs. These types of models are most commonly used for regression, where the target is a continuous variable, or classification, where the target is a categorical variable. When building and evaluating a supervised learning model, the data is generally broken into two partitions: the training data set (70-80% of the observations) and the test (or validation) data set. Models are built on the training data, then their performance is evaluated on the test partition. This is done to prevent overfitting a model on a specific data set and to see how the algorithm will perform on new observations. Let's look at some of the more common types of supervised learning models you may encounter when doing clinical research.

TYPES OF SUPERVISED MODELS

LOGISTIC REGRESSION

There is some debate over whether a logistic regression is a purely statistical model or a machine learning model, but it is generally grouped with ML models. Logistic regression is generally used to predict binary outcomes (Yes/No, Success/Failure), but can also be used for a categorical variable with multiple levels. This is the first classification model most people learn. It's very similar to a regular linear regression, but there are some differences because you are predicting a binary outcome. Because this type of model is statistically based, you will see things like p-values on different variables to say if they are useful to the model. Logistic regression models can also consider variable interactions. If the inputs are categorical variables, you may have to use 'dummy' variables to code them, otherwise logistic regression will treat them as numeric fields. Logistic regression models produce an equation that can be used to score new observations. The output prediction is a probability – you must specify a prediction cutoff point to classify a new observation as a success or failure (typically the default cutoff point is at .5, so if the model returns a .48, the observation will be classified as a 0). The cutoff point can be adjusted based on how bad a wrong answer is – usually a false negative on a diagnostic test is much more dangerous than a false positive, which might lead to a more invasive test, so you should set the cutoff point to minimize the false negatives. Logistic regression models are also "case complete" models, meaning that if an observation has one missing value out of all the variables, the entire case is excluded from the model. Because of this, imputation of missing values is often performed prior to modelling.

DECISION TREES

Decision trees can be used for both classification and regression (both categorical and numeric targets) but are more frequently used for classification models. They recursively look through all the values of the variables to determine the variable and cut point which will divide the observations into the two most distinct groups. For example, with the SASHELP.HEART data (Figure 1), the overall survival rate of the population is 61.8%. The decision tree looks through all the variables and determines that the best place to have initially split the data is on the "Age at Start" at the value of 48 – this was the value that had the biggest impact on splitting the observations down different **branches** of the tree. Nodes with colors in them represent **leaves**, which are groups that don't need to be split any further. In this example, the yellow box at the top only has 22% survivors (Age >= 56, there were two splits of Age to get to that leaf), while the blue box at the top left has a survival rate of 82% (Age < 48 or missing; BP Normal, Optimal, or missing).

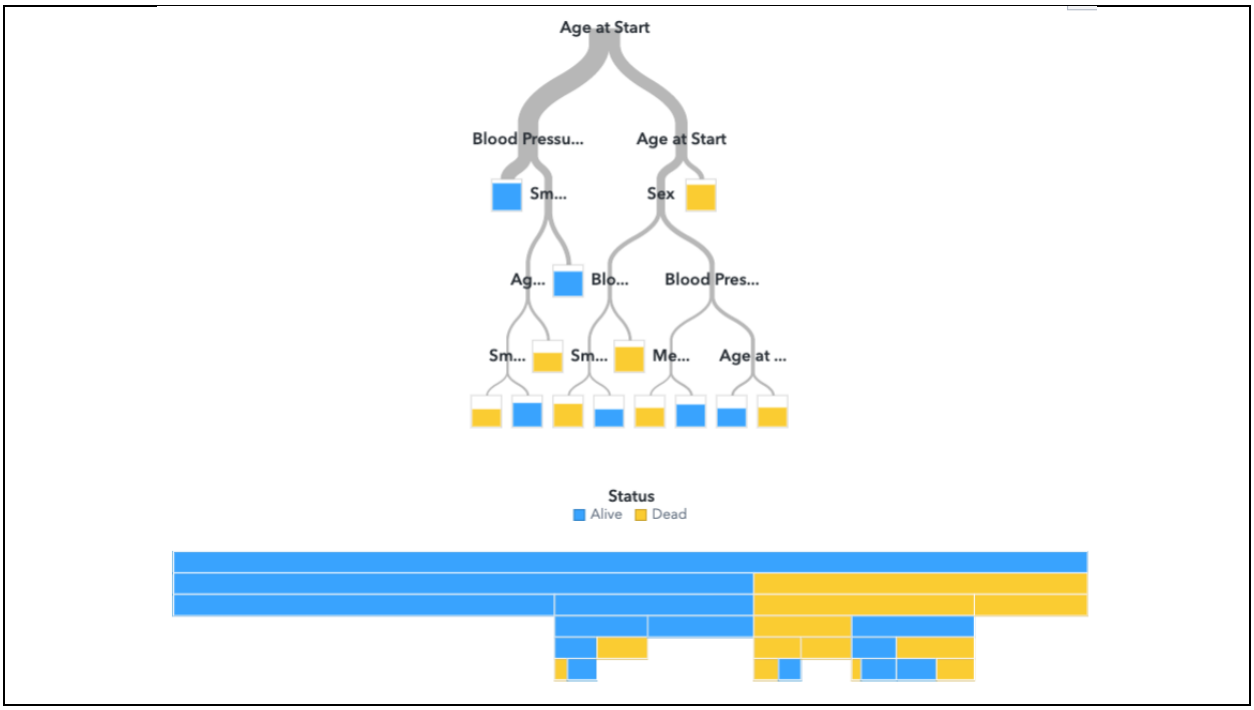


Figure 1. Decision Tree for predicting Status (Alive/Dead) in the SASHELP.HEART dataset.

As you can see in that first leaf, decision trees treat missing values as just another value, so all observations are used. Typical parameters about the model you can set are:

- Maximum Branches – how many splits can you make at any node (here it's 2)
- Maximum Levels – how many splits downward you can make (this tree has a depth of 5)
- Leaf size - the minimum number of observations you can have in a leaf

Decision trees are very easy to understand and to implement – the output of a tree produces a series of if-then rules which can be easily programmed into any language. As such, they are an excellent model to use to explain a particular problem.

RANDOM FORESTS

Random forest models, as the name sort of implies, are a collection of randomly generated decision trees. The algorithm basically works as follows:

1. Data is randomly sampled (with replacement) to create multiple subsets of observations (these are called bootstrapped samples).
2. For each sample, a decision tree is trained using a randomly sampled set of variables.
3. Once the trees have been trained, predictions are made running new data through each decision tree. The final prediction is based on a majority vote of the individual trees.

Because they combine predictions of multiple trees, random forest models are generally more accurate and less prone to over-fitting than single decision tree models.

GRADIENT BOOSTING

Gradient boosting also combines several simple models to create a strong predictive one. Here's how it works:

1. Start with a simple model, like a tree, and train it to make predictions.
2. Calculate the errors between the predicted values and the actual values for each observation.
3. Build a new model to predict the errors calculated in step 2.
4. Combine the base model and the error prediction model to make new predictions.
5. Cycle through steps 2-4 until the new error rate stops improving, or until a set number of iterations has been reached.
6. Final version gives final predictions.

Gradient boosting models are popular because they can handle complex relationships between the features and the target variable. They are also effective at handling missing data and cases with outliers.

UNSUPERVISED LEARNING

Unsupervised ML models are techniques where the data is not labeled or classified, so there is no target to predict. These models are used to discover patterns, relationships, and structures in the data. They are used in cases such as customer segmentation, anomaly detection, reducing the number of variables in a model, and recommendation systems.

CLUSTERING OBSERATIONS

Clustering is a popular method for doing customer segmentation based off multiple inputs. It is also used in variable reduction strategies (for example, instead of using dozens of demographic and lab values at baseline, use those values to group subjects into similar types of subjects, and use that group as a single input variable). The model works as follows:

1. Start by picking a number of clusters you want
2. Randomly assign points as the centroid of the clusters
3. Determine the distance from each centroid to each observation
4. Assign each observation to the nearest centroid
5. Find the new centroid of the observations
6. Repeat steps 3-5 until no observations move clusters and the centroids stay constant
7. Evaluate this process for multiple starting numbers of clusters until you find the number that does the best job explaining the variation (there is a statistical method for that)
8. Assign the cluster number as a feature in the dataset and summarize the clusters

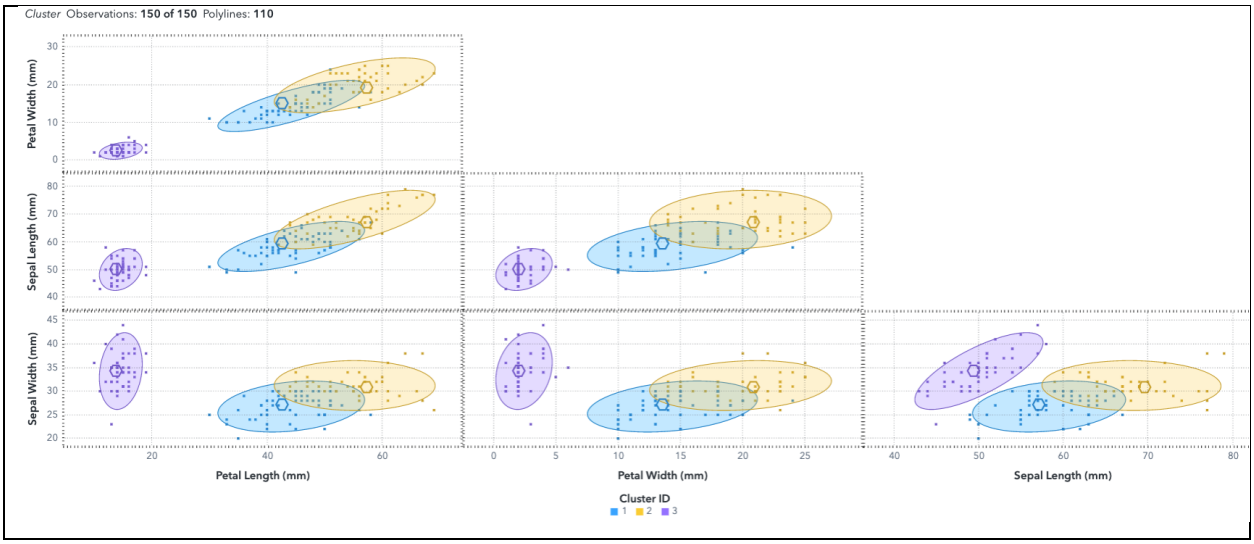


Figure 2. Creating clusters from the SASHELP.IRIS dataset

Figure 2 shows a classic example of clustering data from flower measurements into three groups that have similar measurements of the sepals (the green part that supports the flower) and petals. In this case, cluster membership is an excellent predictor of flower type.

PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) is another technique that tries to reduce the number of features used in a model (also called reducing the dimensionality of a dataset). It determines which features explain the variance in the dataset, then creates a new coordinate system that is defined by these features.

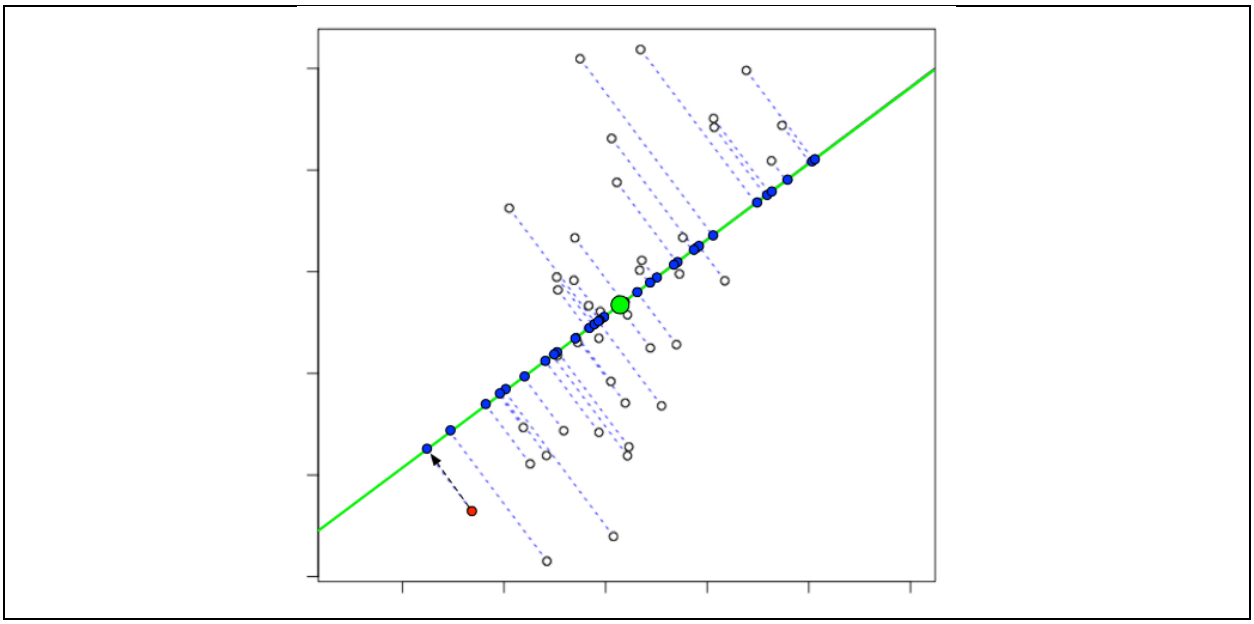


Figure 3. Illustration of reducing dimensionality of a dataset with PCA.

Figure 3 gives a simple example of this – the dataset itself has two features, the X and Y axes. The green line represents a way to reduce those two variables to just one – the position on a new number line explains a large amount of the variability in the two-dimensional data. It gets a little tricky to picture this in your head with large amounts of data, but the algorithm finds the best set of reduced dimensions to explain the data.

1. Standardize each variable in the dataset (subtract the mean and divide by the standard deviation)
2. Calculate the covariance matrix from the standardized data
3. Calculate the eigenvalues and eigenvectors, which are elements in linear algebra. The eigenvectors are the direction of the new axes, the eigenvalues show how much of the overall variance is captured in this direction
4. Choose the principal components – pick the smallest number of eigenvalues that explain most of the variance. They will sum to 100%, usually you pick enough components such that the eigenvalues sum to 90%
5. Project the full data onto the principal components. Now instead of say 15 inputs, you may just have three principal components

PCA is commonly used for compressing image data and feature selection upstream of a new model. Using fewer features can be a big help to ML algorithms that are computationally intensive.

ASSOCIATION ANALYSIS

Association models are used in recommendation systems like those on Amazon that suggest other things you might want to buy once you add an item into your cart or Netflix suggesting your next movie. These models identify patterns in the dataset to see how likely things are to be purchased together. The Apriori algorithm, developed in 1994, is one of the most common used to perform association analysis. Here's how it works:

1. Data is converted into a transactional format – each transaction (purchase, movie rental) has columns for all the items, and a 1 is entered in the column if the item was purchased and a 0 if not
2. Identify sets of items that are frequently obtained together
3. Generate association rules that reflect often people who buy object A also buy object B
4. Rank the association rules by how often items are purchased together

The strongest association rules are then used to make suggestions about what else might be of interest. In addition to market basket analyses applications, association analysis is used in fraud detection and supply chain optimization.

DEEP LEARNING

Deep learning is the branch of ML people most think of when they think about artificial intelligence. As we saw earlier, neural network models have been around since the 1950s, but the explosion of data and computing power has dramatically increased their complexity and usage.

Neural networks got their inspiration (and name) from the structure and function of the human brain, where neurons are connected in complex networks to perform a variety of tasks such as a decision making and recognition. Figure 4 shows the basic layout of a neural network.

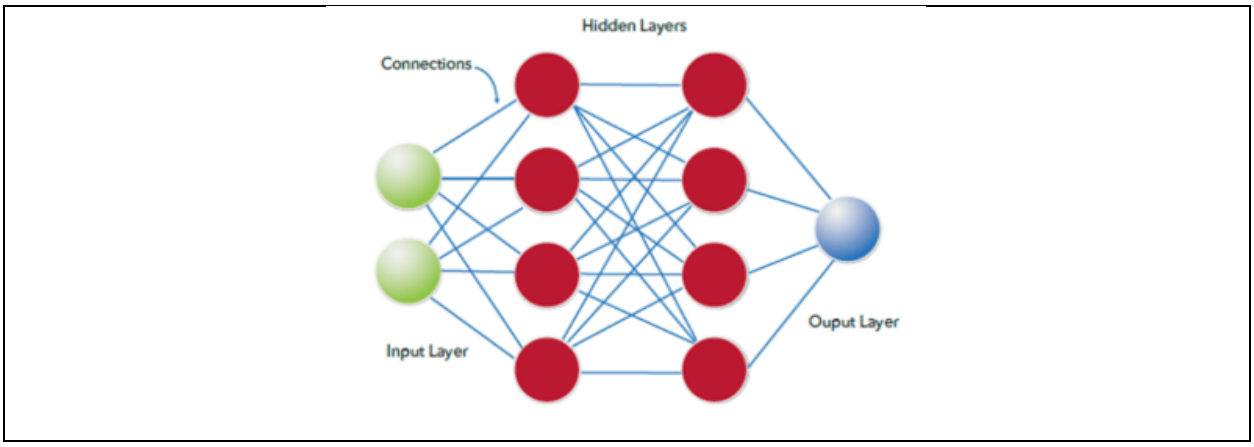


Figure 4. Basic Structure of a Neural Network

Neural networks are made up of three types of layers:

- Input Layer: the data
- Output layer: the value of the target variable
- Hidden layer(s): the intermediate steps to learning how to go from the input data to the target variable

The circles in Figure 4 are referred to nodes. The value at any node is calculated like this:

$$y = f(W * X + b)$$

- **y** is the output of the neuron
- **f** is the activation function that determines the value based on the inputs. There are several different activation functions possible, and some algorithms try several functions to determine the one that makes the best predictions. Different layers can have different activation functions.
- **W** is a set of weights that indicate how important the values of the upstream nodes are.
- **X** is the vector of outputs of the upstream neurons
- **b** is a bias term that makes an adjustment to the output of the activation function.

The training process of the neural network will find the best set of weights and biases to minimize the error between predictions and actual values of the target variable. Even in a simple neural network there can be thousands of parameters to calculate, so these models can be very computationally intense, often requiring GPUs (Graphical Processing Units) to help with the calculations. They also require massive amounts of labeled data. Different types of problems will require different structures (number of hidden layers and nodes in each layer) of neural networks.

- **Convolutional Neural Networks (CNN):** commonly used in computer vision applications
- **Recurrent Neural Networks (RNN):** natural language translation and sentiment analysis
- **Generative Adversarial Networks (GAN):** generate new data that is similar to training data, such as realistic images or videos

MODEL EVALUATION

A key part to training and testing supervised ML and Neural Networks is to use model evaluation techniques to evaluate the accuracy and efficacy of your models. There are many different methods of evaluation; we'll focus on four of the most common.

MISCLASSIFICATION RATE

Recall that the output of a binary prediction model is the probability of having a success (however it was defined). You then apply a cutoff point, and any observation with a probability above that cutoff point has a predicted value of 1 and for the ones below the probability is a 0. The misclassification rate is then simply the percent of occurrences where the predicted value matched the actual value.

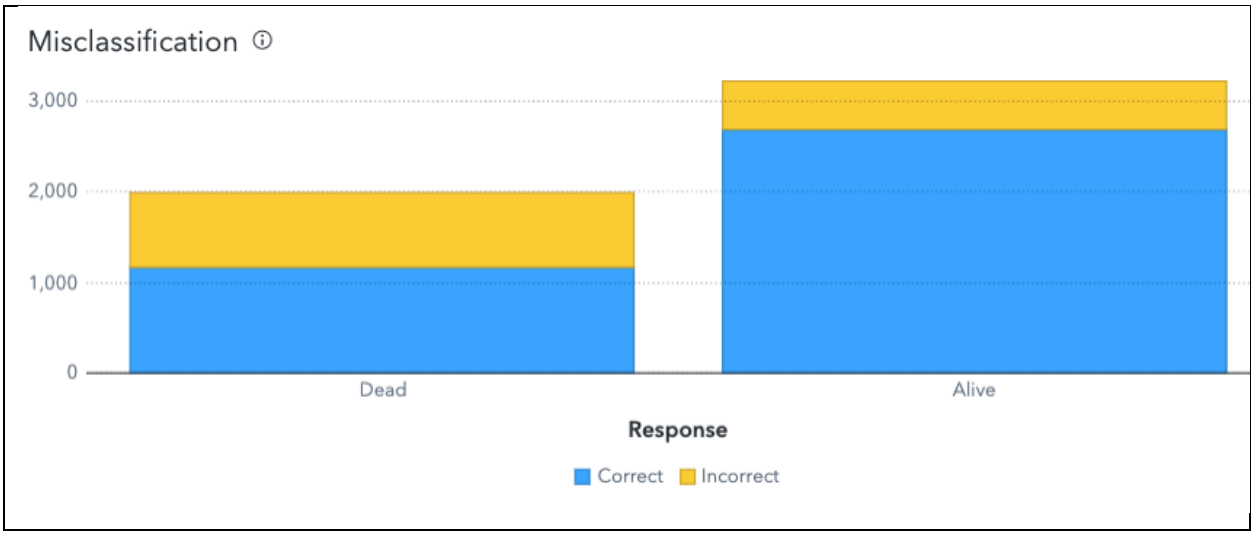


Figure 5. Misclassification Rate

Generally, a lower misclassification rate is better, but you also need to factor in the impact of a false positive versus a false negative.

CONFUSION MATRIX

A confusion matrix is another way of looking at misclassification, except you look at the actual numbers instead of just seeing a misclassification percentage.

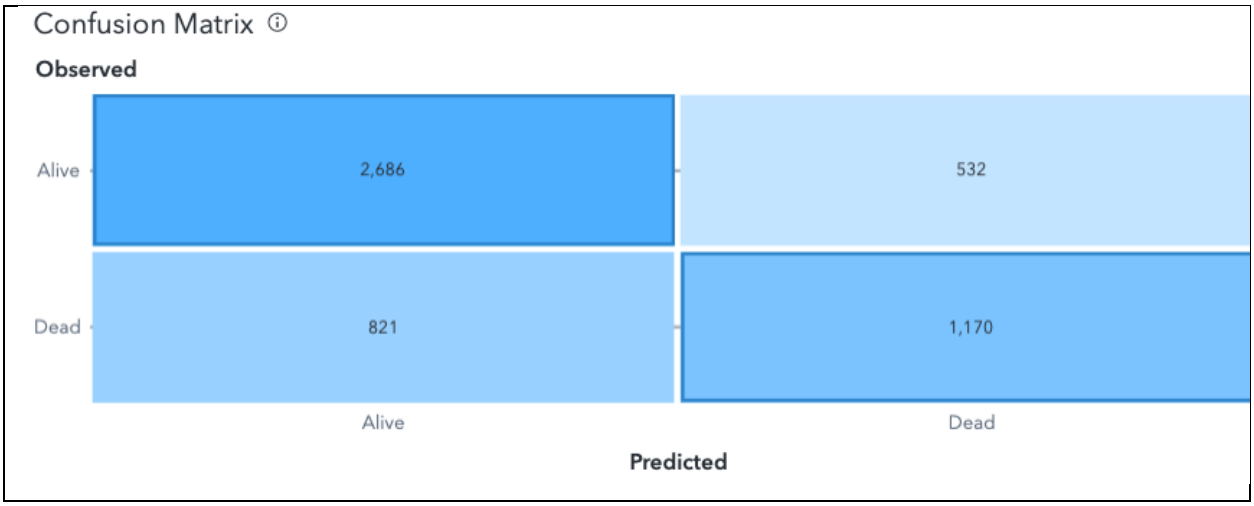


Figure 6. Confusion Matrix

The values in the confusion matrix are used to calculate multiple measures (including misclassification rate). The two most commonly used are:

- Sensitivity – the percent of the time the model predicts an observed success as a success
- Specificity – the percent of time the model predicts an observed failure as a failure

In both the misclassification rate and the confusion matrix, the value of the cutoff point plays a major role. If you change the cutoff point, you change the values in the confusion matrix, and will have different measures of sensitivity and specificity.

ROC CURVES

ROC (Receiver Operating Characteristic) Curves plot the true positive rate (specificity) vs the false positive rate (1 – Specificity) across all possible cutoff points.

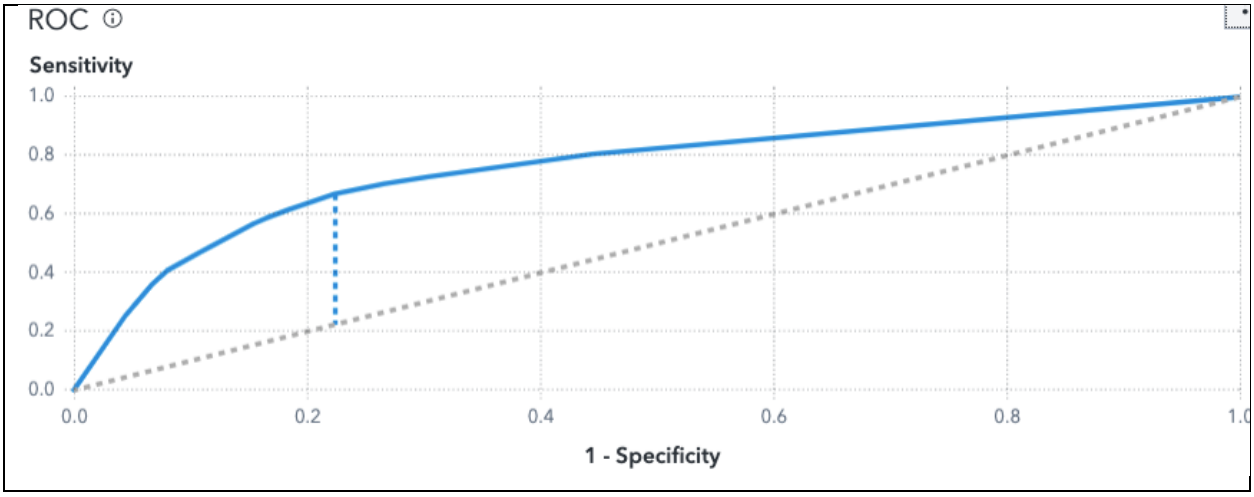


Figure 7. ROC Curve

The closer the curve is to the upper left side of the axis, the better the model is performing. When comparing multiple models, look for the highest curve. Besides a visual examination, the results from the ROC curve are often reported as a single number: AUC, or the area under the ROC curve. For binary outcomes, the AUC is equivalent to the c-statistic, a measure of concordance. These measures all cover every possible cutoff point, so they are good ways to evaluate model performance.

LIFT CURVES

Lift curves show how well the model identifies outcomes compared to random choice. Consider an example where you have a sample of 1000 observations, and you know that 20% of them are classified as failing your test. If you picked 100 of them at random, you'd then expect to see 20 who actually failed the test.

Now, order the 1000 observations by the value of the model prediction, take the highest 100 predictions and see how many of them failed. If you see, for instance, 40 failures, then the model provided a lift of 40/20 = 2.0 for this first group of observations. Move over to the next 100 and do the same thing. A lift curve shows those results.

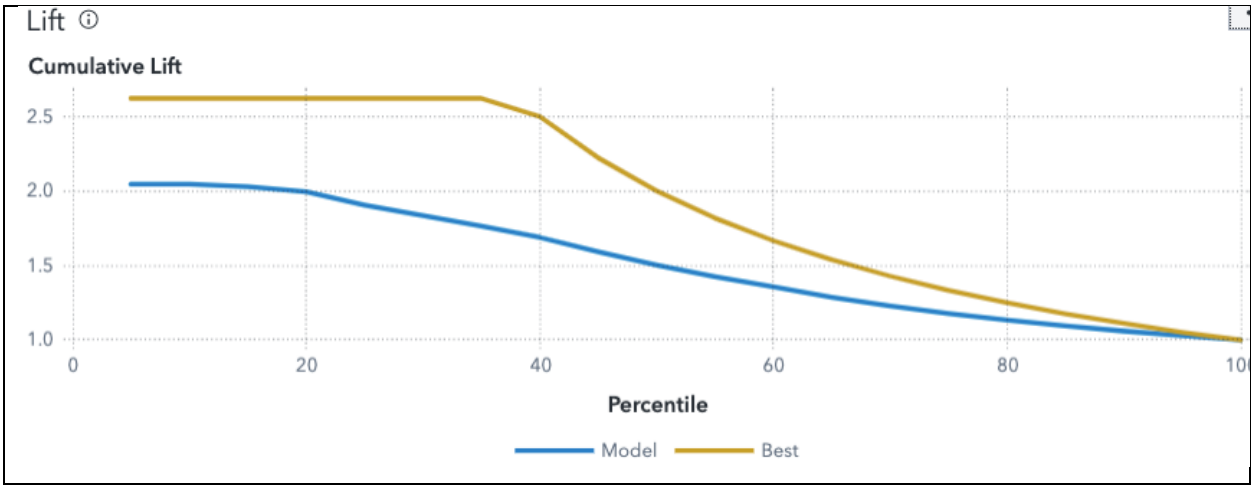


Figure 8. Lift Curve

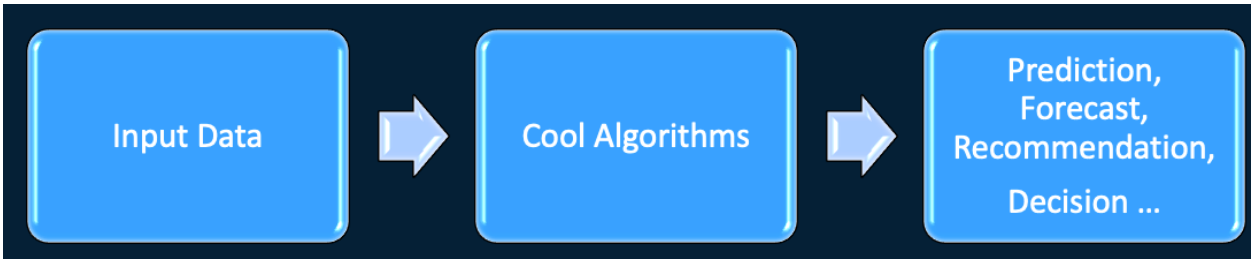
Lift curves will always be over 1, and at the last percentile, will always end up at 1. In Figure 8, the gold line represents the lift for the best possible model, so you can get a sense of how well your model did.

In all these examples, you can compare multiple ML models on the same data to see which one performed the best (which, admittedly, is a bit of a qualitative measure).

IMPLEMENTATION CHALLENGED

There is a strong drive for organizations and governments to implement AI solutions. A primary factor is the cost savings or income generation – these systems may allow companies to better attract and retain customers, or to identify employees who are at risk of leaving. Another factor is that AI systems can five off the impression of impartiality – it's the algorithm that is making the decision about who gets approved for a loan, or who gets a harsher sentence, which can give the impression of impartiality. There's a term for letting the computer take responsibility for making these sorts of decisions – it's called mathwashing. Cathy O'Neil covers this in detail in her book *Weapons of Math Destruction* (2016).

To understand the impacts of bias in machine learning, we'll look at the three steps typically involved with creating an ML application:



BIAS IN TRAINING DATA

The foundation to building a ML model is the training data – this is all the information used by a model to develop later predictions. The most important thing to realize is that **biased data will give you biased results**. It's vital to understand that if you have data collected by humans and/or about humans, then there is bias present in that data, and it's important to identify the ramifications.

One example (of many, many possible examples) details efforts made by Amazon (Dastin, 2018) to build a system to screen through job applicant resumes. The idea was to look at the resumes of all the people Amazon had hired and use that information to rank the job applicants, and to only interview the highest-ranking ones. The problem here was that historically, Amazon had not been hiring women for the engineering roles, and there was an overwhelming percentage of men in the training data. The ML algorithm picked up on that and gave lower scores to any candidate it identified as a woman. The developers were able to institute workarounds to prevent the algorithm from considering specific words (e.g. women, she, her) and correlations like attending an all-women's school, but ultimately the algorithm was so successful at finding ways to exclude women, the team was forces to scrap the system. The algorithm itself did exactly what was asked of it, the problem was it was being trained on data with inherent biases.

The key takeaway here is that models trained on biased data will excel at applying that bias – perhaps even more efficiently than humans. Your responsibility when building or implementing an AI/ML solution is to question the data:

- Where did it come from?
- What historical biases does it include?
- How representative is the data?
- How did it get labeled (why are some records successes and others failures)?
- Is there a feedback loop for addressing problems?

ALGORITHM PERFORMANCE

Machine Learning algorithms do exactly what you ask them to do, which might not be what you meant them to do. There is no nuance or conscience involved, unless specifically programmed in. This may sound obvious, but the implications may be subtle. You may give an algorithm a bunch of labeled pictures of dogs and wolves and train it to tell the difference. Although a person may spend time looking at features such as snout length, coloring, and ear shapes, the algorithm might have an easier time sorting the pictures into mostly correct groups by looking at background image features, like the presence of snow (Ribeiro et al, 2016). It is important to try to understand why an algorithm is making a specific prediction, which can be very trick to ascertain with complex, black-box models.

This can cause real problems, because unless you investigate why a model made the decision it did, you may be implementing decisions based on sub-optimal information. A good example of this was done on a study of chest x-rays used to diagnose cardiomegaly (enlarged heart) by a neural network (Zech, 2018). On first glance, the algorithm seemed to be working fine – it was trained on images that were labeled as yes/no for cardiomegaly, and a validation set was examined. The model performed well, giving high predictive scores to images of patients with the condition and lower scores to those without. Neural networks are complicated, and it is not particularly obvious why the model was scoring the images as it was, but it seemed accurate enough, so it is not hard to imagine it being put into use diagnosing patients. Only when the author of the paper took the effort to use heatmaps to see what parts of the image were most important that he noticed something – the algorithm was looking at parts of the image that had nothing to do with the condition of the patient – specifically, it was looking at some words that were printed on the upper right of the image that indicated the machine used to take the image was a portable x-ray. So, the model was taking into account that the patient was too sick to travel to the imaging room, which is not really information that should be used to make an image-based diagnoses. Essentially, it was cheating a little by knowing the answer before looking at the data. For someone looking at just the model results, though, it would have seemed like an excellent model doing its job.

The key thing to remember is that models are lazy, but effective. They will perform the task without considering context, unless specifically programmed to do so. Your responsibility when building or implementing an AI/ML solution is to question the results:

- Why did the model make this specific prediction for this specific case?
- What are the key inputs being used to make predictions?
- What, exactly, was the model set up to do?

APPLYING THE MODEL TO NEW DATA

The third area of concern is how a model will be applied to new data. Models are only accurate on data similar to what they were trained on, even though they will provide results either way. There can be serious problems with applying a model to novel data. An article int eh Guardian (Devlin, 2018) outlines a relevant concern. A large genetics database was set up in the UK, and machine learning models were used to develop risk scores to predict the onset of schizophrenia. The test produced scores 10 times higher for people of African ancestry than those without. This happened not because there was actually a higher risk of disease, but because the genetics database was built almost entirely upon markers collected from people of Northern European descent. Because the model did not have enough training data on patients of diverse background, the results were wildly inaccurate. Problems of this nature can be difficult to notice, because the model will give results that seem to be based on science, but have little actual value. Your responsibility when building or implementing an AI/ML solution is to question the application:

- Is this data similar to what the model was trained on?
- Do different population subgroups get different predictive results?
- Is there a human feedback loop that allows for training the model?

CONCLUSION

As practitioners of Machine Learning, it is our responsibility to understand how biases can impact the types of models we build and implement. It is altogether too easy to fall into the trap of thinking that because an algorithm gave a result, we eliminated the impact of humans on the decision. The main points to remember are:

- Models can pick up patterns in the data we are not explicitly trying to teach them.
- There is a lack of awareness by most data scientists and statisticians about how historical and societal biases may be present in different aspects of data modeling, including:
    - How we collect and classify data
    - The problems we are trying to solve with ML
    - The Data we choose to train models on and apply results to
    - How we asses accuracy
    - How we present and implement results

REFERENCES

Dastin, Jeffrey. “Amazon scraps secret AI recruiting tool that showed bias against women.” Reuters, October 9, 2018. Available https://www.reuters.com/article/us-amazon-com-jobsautomation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-againstwomen-iduskcn1mk08g

Devlin, Hannah. “Genetics research ‘Biased towards studying white Europeans.’” The Guardian, October 8, 2018. Available https://www.theguardian.com/science/2018/oct/08/genetics-research-biased-towardsstudying-white-europeans

O’Neil, Cathy. September 6, 2016. Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy. Crown Publishing.

Zech, John. “What are radiological deep models actually learning?” Medium, July 8, 2018. Available https://medium.com/@jrzech/what-are-radiological-deep-learning

ACKNOWLEDGMENTS

This paper and presentation are based on work I did with two excellent colleagues: Elena Snavely, Manager of Corporate Analytics & Insights at SAS and Hiwot Tesfaye, Technical Advisor for the Office of Responsible AI a t Microsoft.

RECOMMENDED READING

- Some useful wiki pages about specific algorithms:
  - *https://en.wikipedia.org/wiki/Logic_Theorist*
  - *https://en.wikipedia.org/wiki/Mycin*
  - *https://en.wikipedia.org/wiki/Perceptron*
  - *https://en.wikipedia.org/wiki/Apriori_algorithm*
- A 2020 SAS Global Forum paper on Bias in Machine Learning: *https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2020/4506-2020.pdf*
- The 3Blue1Brown YouTube channel has many excellent videos about ML and math in general. This one in specific does a great job explaining Neural Networks: *https://www.youtube.com/watch?v=aircAruvnKk*
- *https://www.aiweirdness.com/* is a great website about ways AI systems do unexpected things. Her book, **You Look Like a Thing and I Love You** is probably the best introduction to AI you could read.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jim Box
SAS Institute
Jim.Box@sas.com