The Machine Learning Audit—CRISP-DM Framework

Machine learning is revolutionizing many industries, from banking to manufacturing to social media. This mathematical optimization technique is being used to identify credit card fraud, tag individuals in photos and increase e-commerce sales by recommending products. Machine learning can be summarized as a computer recognizing patterns without explicit programming. For example, in traditional software engineering, the computer must explicitly be programmed via control statements (e.g., if this event happens, then do this), necessitating that the engineer design and implement the series of steps the computer will perform to complete the given task. However, when dealing with mass amounts of correlated data (two or more variables moving together or away from each other, e.g., the relationship between temperature and humidity), human intuition breaks down. With advances in computing power, the abundance of data storage and recent advances in algorithm design, machine learning is increasingly being utilized by corporations to optimize existing operations and add new services, giving forward-thinking, innovative companies a durable competitive advantage. This increased usage helps establish the need for machine learning audits. However, a standard procedure for how to perform a machine learning audit has yet to be created. Using the Cross Industry Standard Process for Data Mining (CRISP-DM) framework may be a viable audit solution.

The Machine Learning Audit

There are many possible ways to approach a machine learning audit, ranging from a standard software development life cycle (SDLC) approach to a full code review with mathematical assumption inquiries. As in many areas in life, the Pareto principle, also known as the 80/20 principle, can be applied to show that 80 percent of the benefit is yielded by 20 percent of the work. For example, auditors can provide a high level of assurance on the efficacy of a specific machine learning algorithm by examining a modified, domain-specific version of the SDLC of planning, defining, designing, building, testing and deployment, assuming a traditional waterfall approach. In many cases, the maturity level of data science work flows is below that of traditional software engineers, but a general process should still be covered in their analysis. The CRISP-DM model, arguably the industry standard for how machine learning is conducted by practitioners (even if they have not explicitly followed the framework), follows the same principles, but is modified to the needs of the machine learning process. The steps are:

1. Gain an understanding of the business
2. Gain an understanding of the data
3. Prepare the data

Andrew Clark
Is a principal machine learning auditor for Capital One. At Capital One, he establishes approaches for auditing machine learning algorithms and prototypes ways to use machine learning for audit process optimization. He has designed, built and deployed a continuous auditing infrastructure across 17 global manufacturing subsidiaries for a publicly traded manufacturing conglomerate and built a consolidated data mart off the American Institute of Certified Public Accountants Audit Data Standards.
4. Complete modeling
5. Evaluate
6. Deploy

By following the CRISP-DM approach, a level of assurance can be obtained by a high-level review, with more assurance provided if subject matter experts examine each step in more depth. It is important to note that the CRISP-DM framework needs to be modified for more targeted auditing and is more crucial than ensuring that the proper steps of work are documented. That is the reason for proposing the CRISP-DM. For the data preparation, modeling and evaluation stages, a thorough evaluation usually requires a significant knowledge of programming, databases, linear algebra, probability theory, statistics and calculus, but, arguably, one of the most important steps in a machine learning audit can be conducted without these skills, assuming the assistance of the audited, by relying on the traditional auditing technique of sampling.

In the machine learning community, there is considerable interest in making interpretable machine learning models, where individuals can understand that a given classification was made (e.g., this radiology report shows cancer, vs. a noncancerous growth) purposefully. This is important since, in many domains such as medicine, individuals will not trust an algorithmic result unless he/she understands why the prediction was made. Several frameworks exist for auditing machine learning algorithms, such as the Local Interpretable Model-Agnostic Explanations (LIME)\textsuperscript{7} and FairML\textsuperscript{8} frameworks; however, these frameworks provide only an interpretation of the model weights and not a risk-based holistic understanding of the machine learning process. This is where the CRISP-DM approach comes into effect. It should be noted that the LIME and FairML frameworks can be utilized in conjunction with the CRISP-DM framework in the evaluation stage to assist the auditor in understanding the model.

When a machine learning model is fully trained and put into production, it receives data from one set of attributes at a time or as a data stream, depending on the use case. For this example, assume a discrete model with a single set of attributes given to the model one at a time. In either case, after examining what the input parameters are, the auditor could derive a pseudo set of data to feed into the algorithm and examine the predicted outcomes for characteristics that would help to expose any potential biases, e.g., a loan prediction model discriminating against a racial group by zip code alone. By feeding in data over a gamut of possibilities, assurance over the potential biases of the performance of a model can be obtained without fully explaining how or why the algorithm makes a certain prediction. Even with a subject matter expert, a globally interpretable model evaluation method does not currently exist for certain models (e.g., support vector machines and neural networks). By evaluating the output of this sampling series (which should be repeated multiple times with the same input data to ensure consistency), practical accuracy can be determined compared to the mathematical accuracy used when models are being trained by data scientists (the performance, i.e., the accuracy of the model and its ability to meet the business requirements are less complex to ascertain).

“IN THE MACHINE LEARNING COMMUNITY, THERE IS CONSIDERABLE INTEREST IN MAKING INTERPRETABLE MACHINE LEARNING MODELS, WHERE INDIVIDUALS CAN UNDERSTAND THAT A GIVEN CLASSIFICATION WAS MADE PURPOSEFULLY.”

Business Understanding

The business understanding section should be relatively straightforward from an audit perspective, but can be challenging during the development of a model. This section addresses what the business use case is and, with the help of a domain expert, what attributes of the use case should be included in the model such as income amount, job title, education level. In sophisticated environments, when other types of models have already been used, either as software or mental decision models, this step can be a little easier than starting from
As the CRISP-DM framework is iterative, the business understanding section will be revisited often during a large project.

**Data Understanding**

Data understanding is an incredibly important step of the CRISP-DM framework. Without understanding the nature and idiosyncrasy of the data, an accurate model cannot be constructed. However, there is more to this step than meets the eye, since most data, besides categorical variables, have an inherent scale to them, such as Celsius, Fahrenheit, kilometers, miles, etc.

Another important consideration is where the data are housed. Different stores of data have different considerations such as the given schema of a relational database. Without a thorough understanding of the data, strong algorithmic models and their subsequent audits cannot be accomplished.

```
WITHOUT A THOROUGH UNDERSTANDING OF THE DATA, STRONG ALGORITHMIC MODELS AND THEIR SUBSEQUENT AUDITS CANNOT BE ACCOMPLISHED.
```

The auditor needs to be vigilant at this stage to understand all the variables and ensure that these variables are not in conflict or introducing biases. Correlation and covariance matrices could be examined at this stage to understand how the variables correlate and vary in response to one another.

**Data Preparation**

Once the data scientist understands what the use case is, how the data have been collected and other details, preprocessing the data into a usable form for modeling is required. For relational data, there may not be much “wrangling” required to get the data into an amendable structure. However, with unstructured text, such as log files and website scrapped data, the preprocessing stage can be very time consuming. Techniques such as regular expressions (regex) may be required to separate text strings such as extracting an IP address from a log file. The following regex command is able to parse out an IP address, for example:

```
\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b
```

In most instances, the data need to be scaled so that all the features or dimensions have the same scale. Usually z-score standardization is used, yielding a mean $x=0$ and a standard deviation $s=1$ of:

$$z = \frac{x - \bar{x}}{s}$$

For instance, continuing with the Celsius vs. Fahrenheit example, using a set of Celsius values, $C = \{10, 30, 25, 15\}$ and a set of Fahrenheit values $F = \{80, 37, 52, 47\}$, one can scale them by calculating their means:

$$\bar{x} = \frac{x_1 + x_2 + x_3 + x_4}{n}$$

and their standard deviations:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

After standardization, the Celsius and Fahrenheit scores are: $C_z = \{-1.1, 1.1, 0.5, -0.5\}$ and $F_z = \{1.4, -0.9, -0.1, -0.4\}$. When the data are standardized, the individual scales are removed, i.e., if the Fahrenheit values are first converted to Celsius and then standardized, the same result will be achieved.

**Modeling**

Modeling is the vital component of machine learning. It is commonly assumed that data scientists and machine learning engineers spend much of their time modeling; however, in most machine learning projects, modeling is one of the shorter steps, at least for the initial implementation. Many different knobs can be tweaked in different directions to refine the performance of an
algorithm. However, many data scientists use intuition followed by brute-force grid search techniques in practice to try out all the available hyper-parameters (parameters set prior to training that are not learned, within a given range of values, yielding the best output). Depending on the number of hyper-parameters tried and the complexity of the algorithm, this can be a very computationally intensive task. If algorithms have not been tuned, then the models are most likely not fully optimized. This usually implies that they have not reached their global minima, but a lack of model tuning does not endanger an algorithm’s understandability. See figure 1 for a visual representation of the difference between a local and global minimum.

Recent developments in the field of automated machine learning are increasingly used not only to tune the hyper-parameters of models, but to select the specific algorithm itself. When using automated machine learning, the data scientist and auditor need to be vigilant and examine the model selected to ensure that the degree of interpretability required for the given use case is met. This means that the business must explain why each decision was made, as in the case of companies’ subject to the “right to explanation” clause of the European Union’s (EU) General Data Protection Regulation (GDPR). In this context, a nonlinear support vector machine model would not be an acceptable choice. One clear benefit that the GDPR has influenced is more emphasis on model interpretability in algorithmic design. In 2016, the International Conference on Machine Learning (ICML) began a yearly workshop focusing on model interpretability, aptly called the “Workshop on Human Interpretability (WHI) in Machine Learning.”

An extremely important machine learning audit checklist item should be examining if the data were bifurcated into training and test sets. Splitting apart the data helps to prevent against model overfitting, which means the algorithm matches the characteristics of the individual data set too closely, causing it not to generalize well to new data. Traditionally, data are split in an 80/20 split, 80 percent of the data as training data and 20 percent as testing data. Modern best practices take this a step further, using a cross-validation process to split the training data into smaller pieces and testing the model on random subsets of the data. A common approach used is called k-fold cross-validation. This divides the data set into k-1 folds (the k is specified by the machine learning engineer), where the data are iteratively split and tested against the held-out fold to reduce the risk of over-fitting the model to a particular data set. Figure 2 illustrates when a model is underfit to the data, correctly fit to the data and overfit to the data, which has the lowest error, but will not predict new data well.
Evaluation

The evaluation section is arguably the most important section from an audit perspective. It is in this area of the machine learning process, or pipeline, that the model is validated for accuracy and the individual weights can be seen for some models. Traditionally, models are evaluated on their prediction accuracy and generalizability to the production data. However, from an audit perspective, an evaluation of the outcome is a key concern. If the model has an extremely high prediction accuracy (90 percent) and appears to be generalizing well, it still may not be meeting the goals of the model and/or be violating the principles of the business, such as inadvertently performing racial discrimination. In addition to examining all the steps outlined so far, the auditor should create a sample set of data to feed into the algorithm and evaluate the outcome to look for any unintended effects the model may produce. For example, for a loan approval model, the auditor could create a data set with zip codes from affluent, middle class and poor neighborhoods; 90 percent income, median household and poverty level; and each major ethnicity in the region, Caucasian, Hispanic and African American, with each combination of the above, producing a total of 84 data points.

More variables may be in the model, such as credit score, employment information, etc. Test variables would need to be created for these as well. The auditor may discover a culture bias in the model, which may increase the accuracy of the model, but enforce a self-perpetuating bias. This, in turn, could lead to bad publicity and the decrease in accuracy of taking out the race variable; for instance, it may increase the revenue and profit generated from the model. Of course, this is a simplified example and many models may not have any social issues involved, but the process for identifying potential trouble spots and testing for them remains.

Deployment

Specifically, how and where the algorithm in question is deployed is less of a concern to the auditor if the service level and desired capabilities are being met. However, there is one area from which auditor awareness and examination could provide value: technical debt. Whenever a developer is building a system, certain decisions will be made about language, application programming interface (API), open-source libraries, how much documentation to create, how many unit tests to create, etc. Essentially, technical debt is less-than-ideal factors integrated into a system. Technical debt is not inherently bad. It is the result of decisions made to get projects done on time and within budget. However, it is not without consequences. In machine learning, technical debt is harder to spot and remediate than in traditional software engineering projects because of the learning aspect. For the purposes of this example, the focus is on correction cascades, an insidious variety of technical debt. A correction cascade occurs when the algorithm is not producing the desired result and rule-based “fixes” are applied on top of the model to correct for its deficiencies.

Figure 2—Overfitting

These deficiencies might be outlier cases or have occurred because of a poorly trained model or inadequate training/validation data. The problem is that if too many fixes are applied when the model is being trained and tweaked, it becomes increasingly difficult to ascertain what changes to the model will produce improvements, since filters are on top of the results and essentially create an upper bound on the learning capability of the model. Technical debt can be spotted by an experienced, knowledgeable data scientist working on the model. However, the knowledge gained from an audit report may solidify the need to retool a model that data scientists already knew had technical debt.

Conclusion
The CRISP-DM framework has been introduced to instruct auditors on how to perform a high-level machine learning audit. For a particularly deep dive, a machine learning specialist will be required, but by following the given framework, machine learning auditing can be accessible to more audit departments.

Endnotes
7 Ribeiro, M. T.; S. Singh; C. Guestrin; “Why Should I Trust You?,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16, 13 August 2016
8 Adebayo, J. A.; “FairML: ToolBox for Diagnosing Bias in Predictive Modeling,” DSpace@MIT, 2016, http://hdl.handle.net/1721.1/108212
12 Op cit, Goodman