Mastering Predictive Coding: The Ultimate Guide

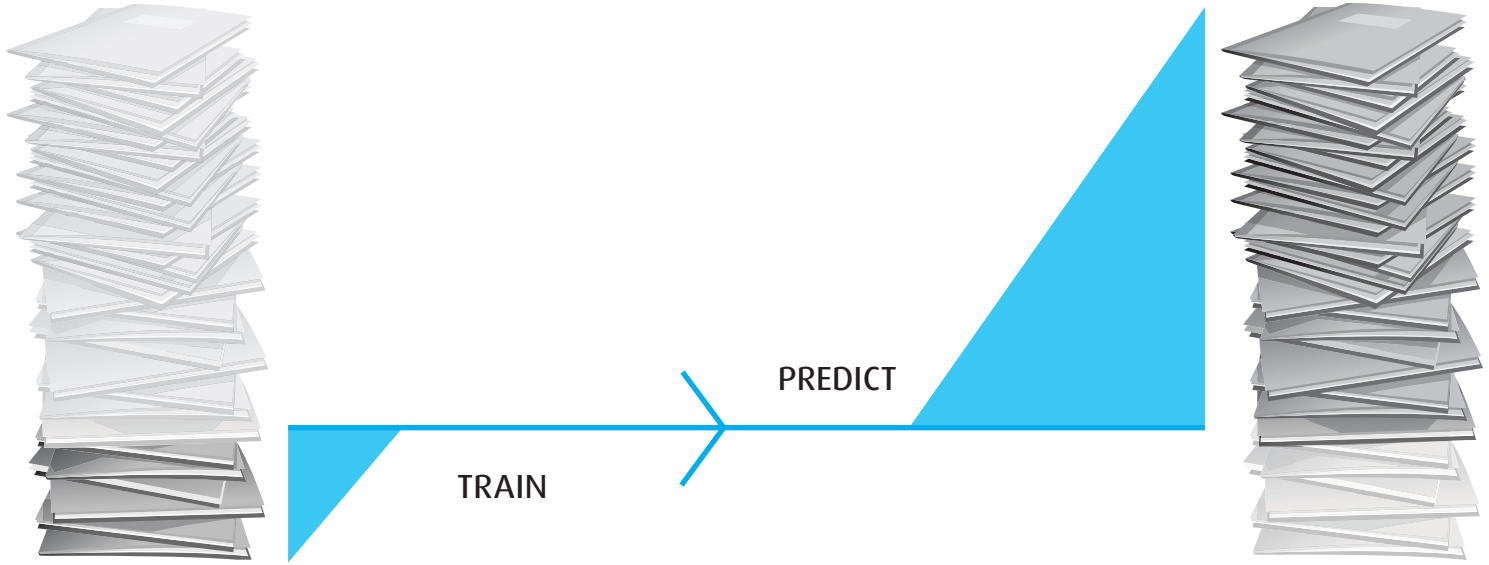
Key considerations and best practices to help you increase ediscovery efficiencies and save money with predictive coding

Table of Contents

1. Leverage Technology, Not Bodies

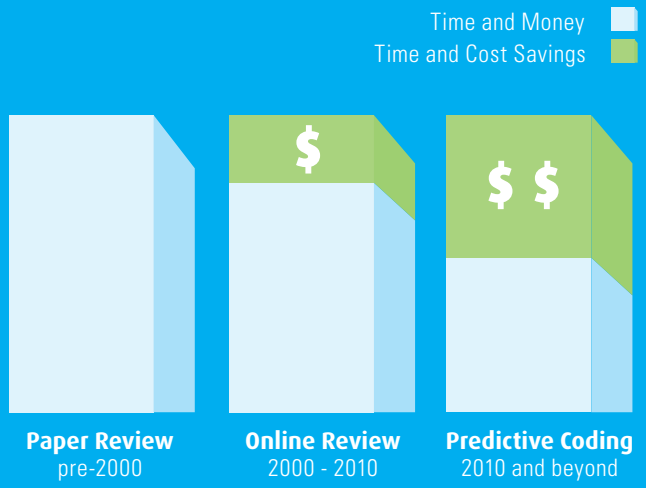
Predictive coding is a document review technology that allows computers to predict particular document classifications (such as “responsive” or “privileged”) based on coding decisions made by human subject matter experts. In the context of electronic discovery, this technology can find key documents faster and with fewer human reviewers, thereby saving hours, days, and potentially weeks of document review.

While predictive coding is a valuable tool for litigation, its adoption, implementation, and management can be challenging for an unprepared organization. This guide seeks to shed light on the obscurities behind the often opaque declarations that predictive coding will "revolutionize ediscovery" by offering the "nuts and bolts" of getting a project off the ground with predictive coding.



2. The Case for Predictive Coding

The review phase of the Electronic Discovery Reference Model (EDRM) is routinely the most expensive part of the ediscovery process. According to the Rand Institute's *Where the Money Goes*, review accounts for roughly 70% of all ediscovery costs — but that also means that review represents the greatest area for potential cost savings. To combat the voluminous cost of review, legal teams started leveraging a variety of review tools, starting with online review in the 2000s and predictive coding in approximately 2010.

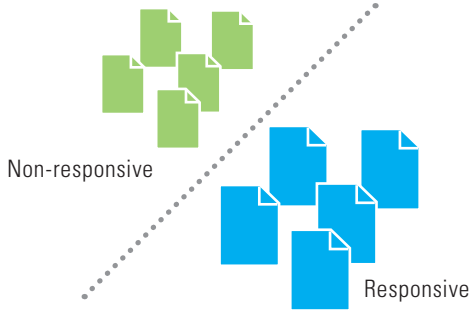


For years, lawyers relied solely on standard keyword searches to recover documents, but many studies suggest that keyword search is not particularly effective. For example, in a 1985 study by David C. Blair and M.E. Maron, the litigation team believed their manual, keyword search retrieved 75% of relevant documents, but further analysis revealed that only 20% of the relevant document pool was retrieved. Similarly, in a 2011 study by Maura R. Grossman and Gordon V. Cormack, recall rates for linear human review were approximately 60%. This is not to say that keyword search is not valuable, but it is very difficult to gather results that are not over- or under-inclusive using keyword search alone.

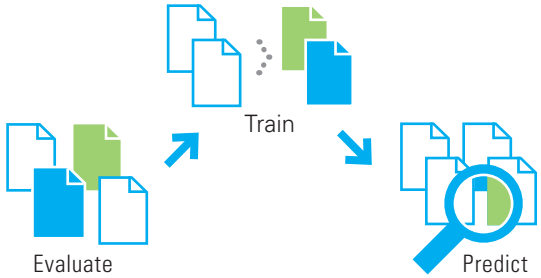
Now, ediscovery professionals can employ predictive coding to bolster the search and document review process, which saves time and costs by helping solve the following key problems:

Finding the right documents as fast as possible

Sorting and grouping documents more efficiently



Validating work performed before production

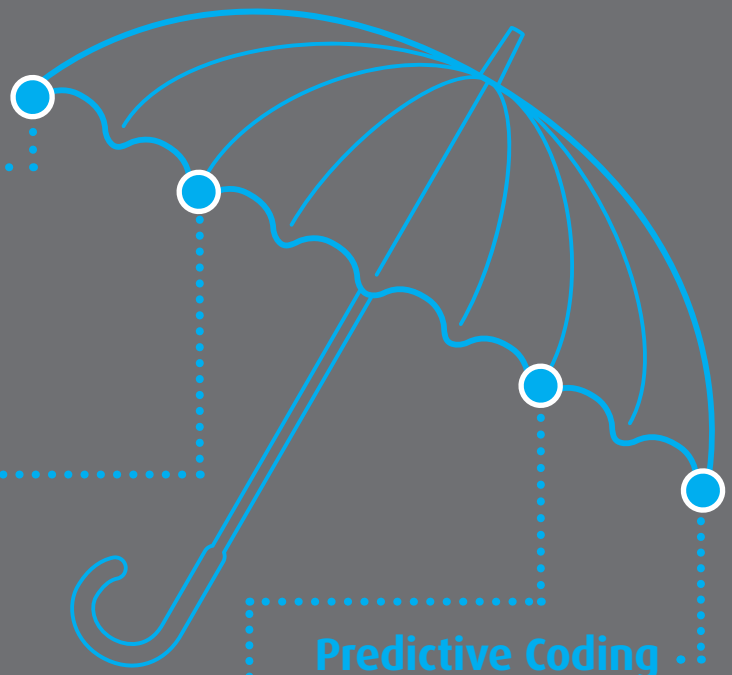


3. Key Terminology

Before diving into predictive coding, it is important to clear up some semantic ambiguity, as widespread adoption of this technology has likely been stifled by confusion about how predictive coding differs from other advanced review technologies. "Predictive coding" goes by many names that are often used interchangeably, such as "computer assisted review" and "technology assisted review."

It is easiest to think of "technology assisted review" (or TAR for short) as an umbrella term encompassing a variety of other review technologies specifically designed to enable reviewers on document analysis teams to work more efficiently and ease the burdens of standard linear review. Under this framework, predictive coding is just one prong of the larger array of complimentary TAR tools.

Defining Technology Assisted Review (TAR)



Visual Analytics

- **Email Threading** chronologically recreates an entire email discussion, adding context to what otherwise would be a disjointed set of email messages and responses.
- **Topic Grouping** uses linguistic algorithms to automatically organize a large collection of documents into thematic groups.

Search

- **Boolean Logic** retrieves documents containing selected search terms. Boolean searches also apply algorithms to identify word combinations, root expanders, and proximities of multiple search terms.
- **Concept Searching** applies algorithms that consider the proximity and frequency of words in relationship to the selected keywords to identify related concepts in a document set.

Predictive Coding

- Categorizes and prioritizes document classifications based on machine learning from coding a seed set of documents.

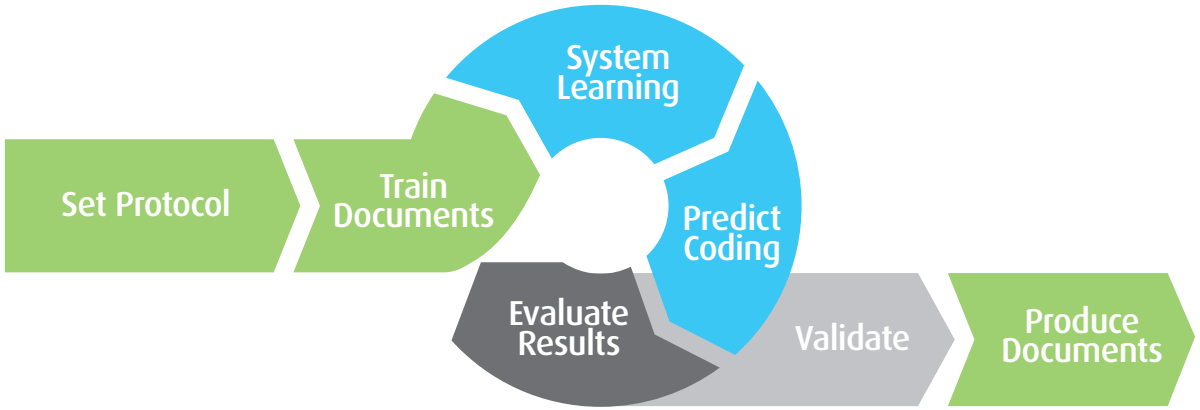
Deduplication

- **Global or Custodian Deduplication** removes identical documents.
- **Near Deduplication** allows reviewers to identify documents that are similar but not exact duplicates to identify discrepancies.

4. The Predictive Coding Process

To best leverage predictive coding, it is important to understand that using this technology is a *process*, most easily broken into six stages.

1. **Set Protocol**
2. **Train Documents**
3. **System Learning / Predict Coding**
4. **Evaluate Results**
5. **Validate**
6. **Produce Documents**



Each stage plays a critical role in properly employing this technology, and each phase brings key considerations to optimize output and maximize efficiency. Here is a brief overview of things to consider:

1. Set Protocol

- Determine discovery goals
- Scope data type and volume
- Consider predictive coding usage
- Plan for production

2. Train Documents

- Choose learning strategy
- Define sample parameters
- Define seed set
- Develop responsiveness standards
- Identify and educate trainers
- Monitor consistency and coding disputes

3. System Learning / Predict Coding

- Coordinate training and learning cycles
- Adjust learning strategy, if needed
- Adapt to new documents

4. Evaluate Results

- Understand effectiveness metrics & statistics
- Address problem files
- Review suggestion results
- Choose next training set
- Continue additional training, if needed

5. Validate

- Perform QC
- Use sampling to validate final document sets
- Confirm completion

6. Produce Documents

- Revisit production specification
- Create production set
- Determine structure of privilege log

The remainder of this section dives into these key considerations and offers practical advice for mastering your predictive coding workflow.

4.1 Setting a Protocol for Predictive Coding

The benefits of predictive coding are only realized when the entire review team aligns strategies at the front-end of the project. Producing satisfactory results using these technologies is highly dependent on proactive planning and maintaining a level of flexibility as the case progresses.

According to U.S. Magistrate Judge Andrew Peck in *Da Silva Moore v. Publicis Groupe,* the goal is not perfection, but rather to create a process that is reasonable and proportional to the matter. Teams tasked with creating a predictive coding protocol should consider the following:

The nature of the case

- Number of documents
- Complexity of the issues
- Number of custodians
- Discovery agreements

Production requirements

- Who are you producing to?
- What do local court rules say about discovery and production?
- How tech-savvy is the adversary or court?
- Will there be rolling productions?

Predictive coding comfort level

- Level of experience with technology
- Reputation of software or service provider
- Solution capability

Internal capabilities

- Budget for predictive coding
- Available review team
- Subject matter expertise
- Time pressure and deadlines

4.2 Training Documents

Setting a Protocol for Training Documents

Amidst the numerous benefits of predictive coding, there is simply no escaping its one "dirty little secret": *the quality of the entire process is only as good as the training it receives.* Without thorough, consistent training by bona fide subject matter experts, the technology will not learn how to properly make coding suggestions, which ultimately defeats the purpose of using this technology in the first place.

Here are four suggested steps and key considerations for developing internal standards to govern the nuances of predictive coding training:

1. Define Content

- What content are you after?
- Create clear, substantive rules to govern training
- Anticipate change while training

The team's ultimate goal is to identify the legal information needed to conduct the review, and then to craft substantive, discernible standards that trainers can rely upon when assessing the responsiveness of a document.

2. Identify Trainers

- Number of trainers needed?
- Will you use internal or external trainers?
- Consider expertise and team orientation

Keep in mind that *bigger is not always better,* as too many trainers can produce more coding inconsistencies and improperly train the system.

3. Educate Trainers

- How will you deliver training?
- Who will conduct the training?
- Set expectations for length, depth, consistency, etc.

Significant diligence is required to ensure that the right trainers—those who are well-informed on the issues and processes—are involved.

4. Arbitrate Consistency

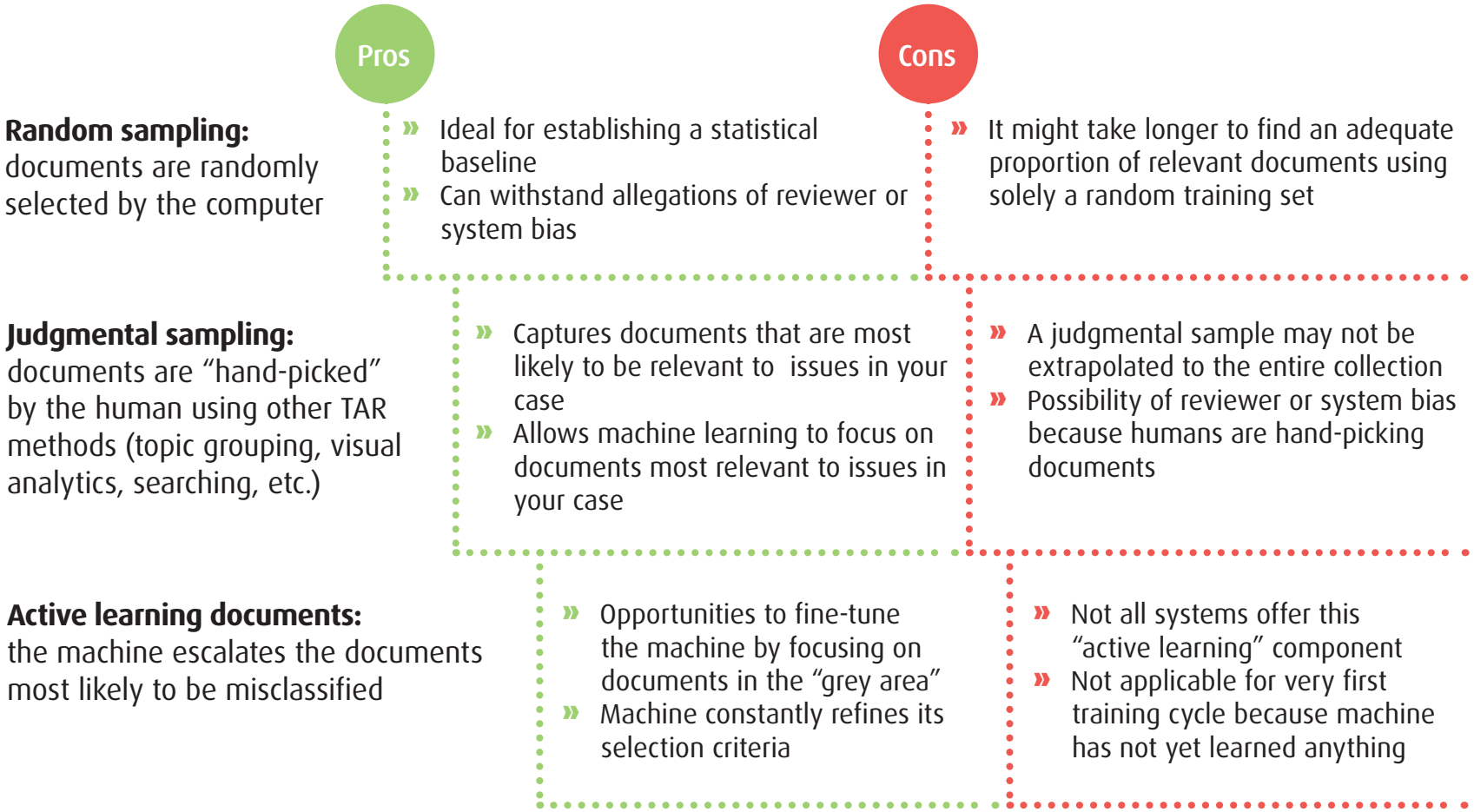
- Create methods for resolving coding disagreements between trainers
- Develop ways to monitor trainer performance

When disputes arise, the team might dedicate a third party to make final decisions on relevance, or encourage trainers to discuss their decisions to reach a resolution.

Creating the Training Set

Once protocols are set, trainers need to look at a fraction of the document set to help the machine learn and determine characteristics of the whole set without having to look at every single document. This subset of documents is referred to as the training set and a review team may select training documents through: (1) **random sampling,** (2) **judgmental sampling**, (3) allowing the machine to **escalate active learning documents** for review, or (4) **some combination of these three methods.** While there is no "optimum" blend of these methods, it is important to recognize the pros and cons of each to devise your strategy.

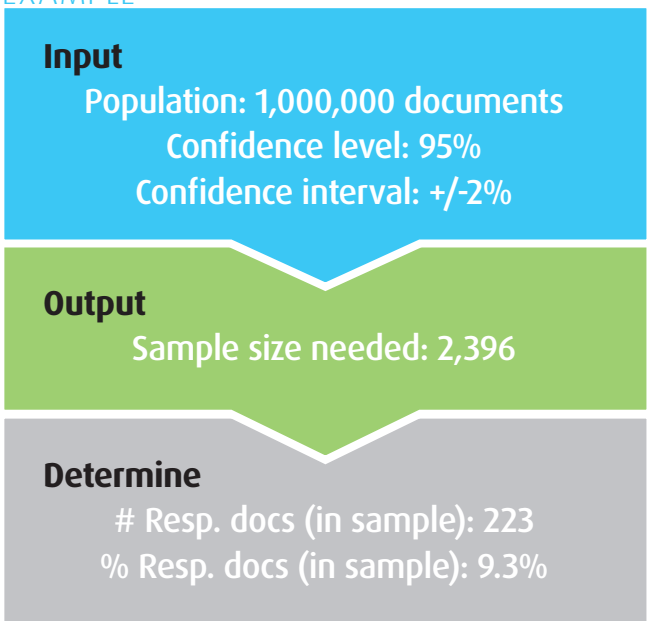
3 Options for Creating a Training Set



Determining the Size of the Testing Set

The size of the sample set is calculated based on: (1) the **size of the document population,** (2) the desired **confidence level,** and (3) an acceptable **confidence interval** (or **margin of error**). In ediscovery, the most frequently used confidence levels and confidence intervals are 95% and +/- 2%, respectively, applied to a population of 1,000,000 documents.

EXAMPLE



Population: The number of documents in the data set.

Confidence level: The probability that the margin of error/confidence interval would contain the true value if the sampling process were repeated frequently. For example, 95% confidence would mean that 95 times out of 100, the responsiveness would be within the confidence interval of +/-2%.

Confidence interval: The range estimated to contain the true value within a specific confidence level. If a sample shows that the proportion of responsive documents is approximately 9%, with a +/- 2% margin of error, this would mean that the responsiveness of the population would range from 7 to 11%.

Sample size: The subset of population used to extrapolate findings to the population.

Plugging the numbers above into a 1,000,000 document population would mean you can be 95% confident that the true proportion of responsive documents is 9% between 70,000 and 110,000 documents (9% is 90,000 documents, plus or minus 2%, or 20,000 documents).

Training is a Process, Not an Event

Predictive coding software learns best from a repetitive cycle of training and learning. Training documents selected in the first instance will be identified in different methods than documents selected for corrective training in later cycles because attorneys naturally learn more about the case as it evolves.

The Trade-offs of the Training Set Size

Generally speaking, larger confidence levels and smaller confidence intervals come at a price. While a higher confidence level is preferable, it either comes at the cost of a larger sample size, which means more eyes placed on a greater number of documents, or a greater range of confidence interval. Similarly, a smaller margin of error desirably reflects more exact estimates, but ultimately comes at the cost of a lower confidence level and/or a larger sample set.

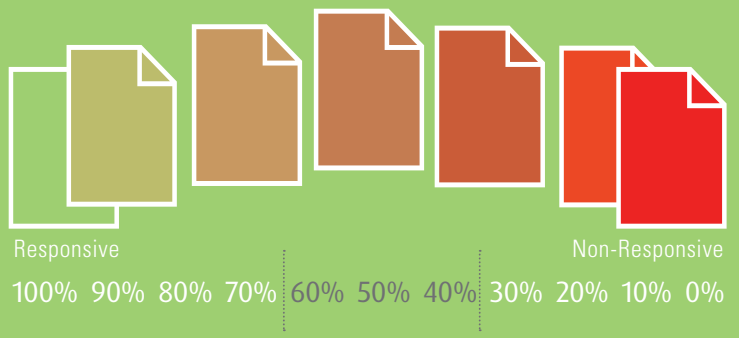
The Bottom Line

Efficient training demands legal professionals who are acutely aware of case details and developments and know how to utilize different types of predictive coding training methods to fit those needs. Keep in mind that no “gold standard” for statistical sampling has emerged, and the prevailing measure is whether the process used was reasonable.

4.3 Understanding Machine Learning and Prediction

Active Learning

In order for the machine to make the most accurate predictions, it must learn from coding decisions made by human reviewers, or "trainers." After initial training, some predictive coding tools offer active learning, where the algorithm starts to route active learning documents that it is uncertain about to human reviewers to determine if they are actually responsive. Unlike random or judgmental sampling, these documents are specifically chosen by the machine and selected from a pool of documents from which the machine needs to learn more about.

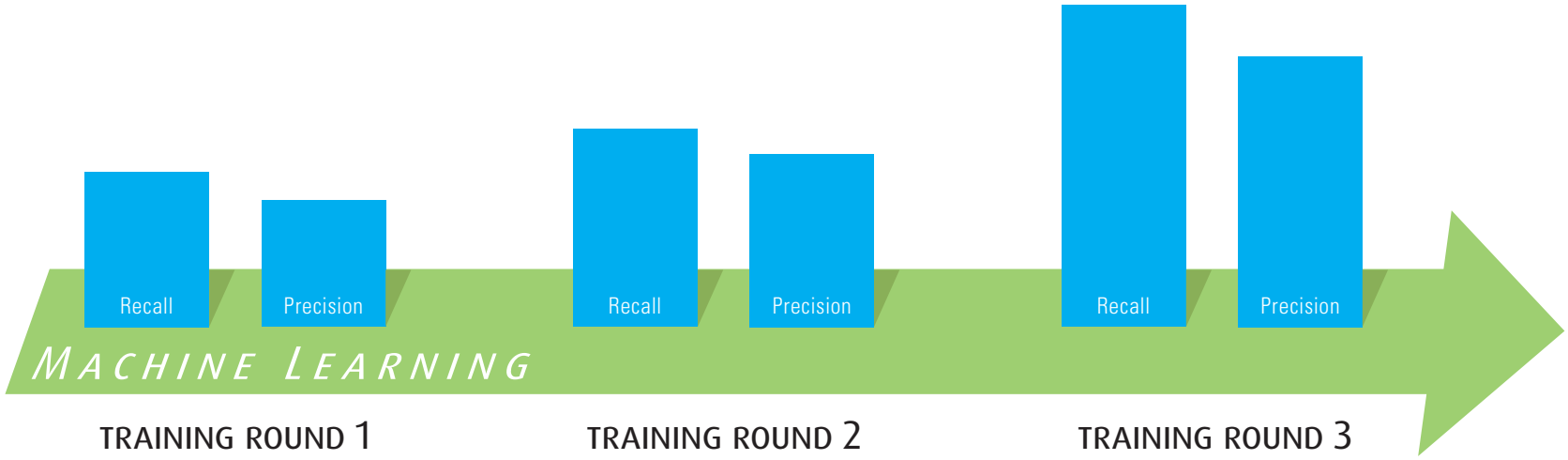


Active Learning Documents come from "grey areas" where the machine cannot identify with certainty whether a document is responsive or not.

How Active Learning Can Make or Break Your Case

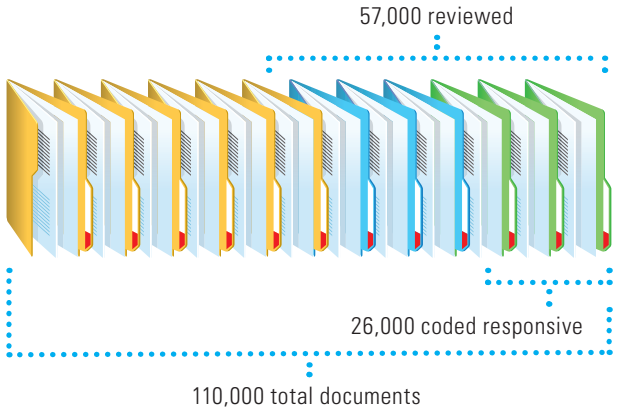
By introducing attorneys to documents on the fringes of relevancy decisions, the machine helps attorneys analyze their case and identify potentially game-changing documents that are difficult to classify. By routing these close calls to human reviewers, the machine allows attorneys to employ their legal analysis to make important decisions about documents that might otherwise be missed via regular search logic.

Active learning is also one of the most effective ways to boost the results that your machine produces in the early rounds of predictive coding training.

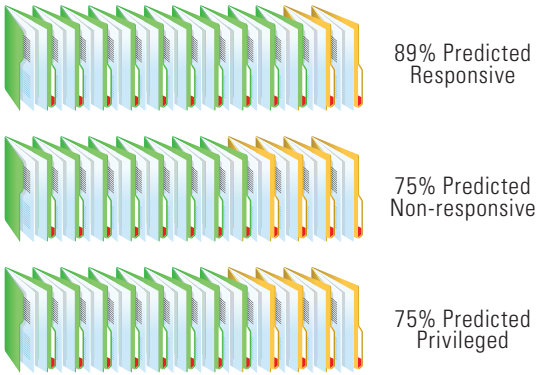


To evaluate the output of your predictive coding tool, you must first understand the different predictive functions it can perform. The training performed by human reviewers creates a predictive model that can be used to either: **(1) rank and prioritize documents based on two relevancy categories or (2) suggest document categorizations across multiple categories.** These two components may be used individually or in tandem: one team may choose to rank the review set and pull only the top ranked documents, while another might train the full predictive coding system and use it to make various coding suggestions. In addition, keep in mind that various predictive coding software platforms have differing capabilities and feature sets when it comes to ranking and suggesting categories.

Prioritization



Categorization



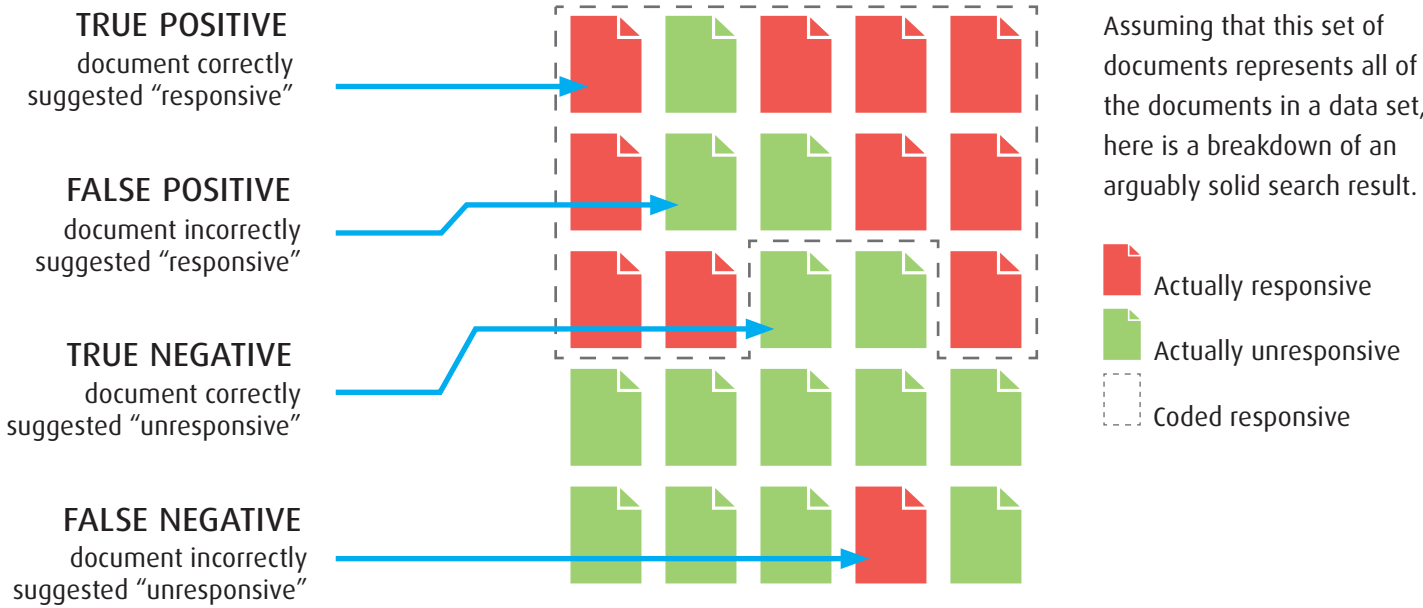
4.4 Evaluating the Results

Multiple Iterations

The training, prediction, and evaluation stages of the predictive coding cycle are an iterative process. If the quality of the output is insufficient, additional training documents can be selected and reviewed to improve the system's performance. Multiple training sets are commonly reviewed and coded until the desired performance levels are achieved.

How to Tell if Additional Training is Necessary: Understanding Metrics

After your team trains the computer and runs the machine learning, the system produces a report with numerous metrics in order to evaluate the effectiveness of the system and determine if additional training and an additional training set are needed to improve the results. In order to fully comprehend this report, there are several key metrics you will need to understand.



Understanding these four items will help you evaluate the recall and precision of your search—two key metrics contained in a predictive coding effectiveness report.

Understanding Metrics: Precision, Recall, F-measure

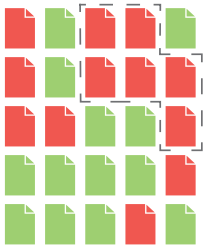
After machine learning, you will get a report that includes the number of true positives/negatives and false positives/negatives, which are used to calculate precision, recall, and f-measure. Understanding what these metrics mean is imperative to determining if you should re-train the system or move forward with the next stage of your review and production.

Precision is a measure of exactness, or the fraction of relevant documents that are actually relevant within the retrieved result

Recall, or the measure of completeness, calculates the number of relevant documents retrieved out of all the relevant documents

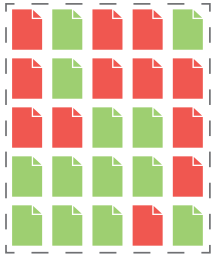
F-measure is the harmonic mean between precision and recall, or the weighted average between those two metrics

1. EXACT, BUT NOT COMPLETE



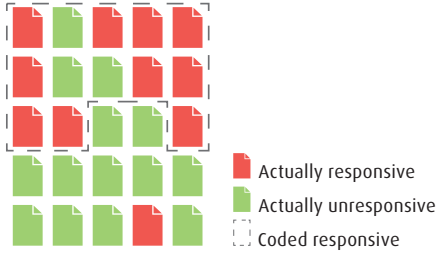
Perfect precision, but low recall. The review team would want to re-train the system to capture more relevant documents.

2. COMPLETE, BUT NOT EXACT



High recall but low precision. The team should re-train to help the machine identify differences between relevant and non-relevant documents.

3. EXACT *and* COMPLETE



These results are arguably good enough to suggest the review team has trained the system adequately.

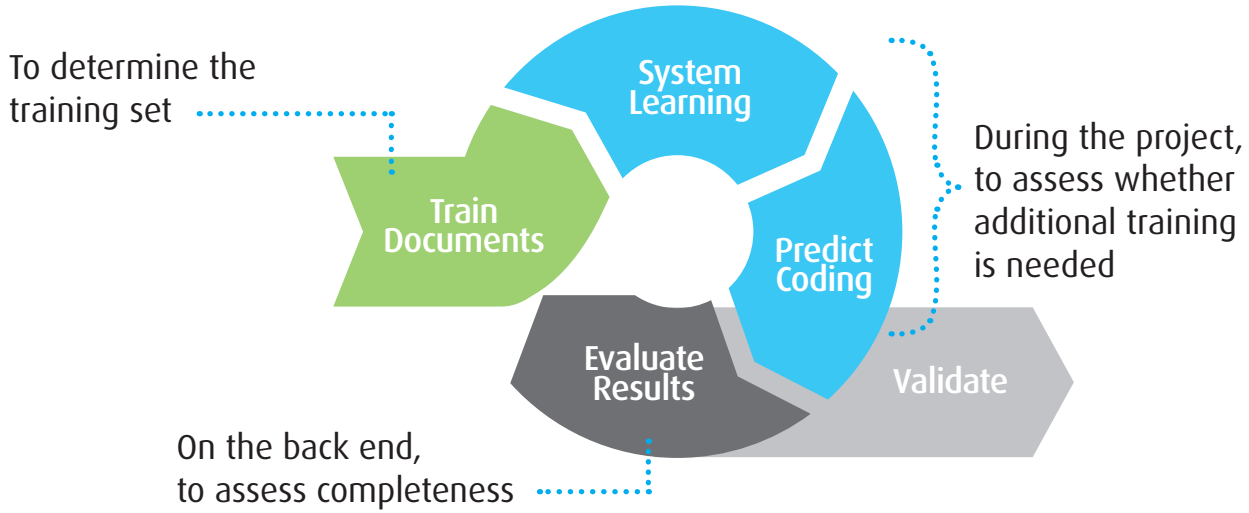
Are Your Metrics Good Enough?

When you evaluate the output from predictive coding, it is important to remember that "good metrics" are relative to the training that has been performed, the case, and your expectations.

The most effective way to evaluate your metrics as part of a reasonable process is to use **sampling,** which rests on a simple premise: *machine predictions are not always right.* The sampling process is most often used to perform quality control by *examining a fraction of the document population to determine characteristics of the whole,* further validating what you do—or do not—have, which strengthens the defensibility of your project.

When to Use Sampling

As one of the most versatile tools available to you, sampling may be performed:



Using Sampling to Get Better Metrics

It is common for categories to underperform after coding the initial training set. If quality control suggests underperforming categories, additional training may be necessary. Essentially, the machine needs to observe decisions on more documents to train and validate the system. Additional training can take multiple approaches:

1. **Train more active learning documents:** Help the machine clarify document classifications it is uncertain about. This ensures that the machine receives an array of examples in the responsiveness spectrum, and will help it define where the cut-off is between responsive and non-responsive.

2. **Perform analysis of machine predictions:** Reviewing the machine's predictions and, if necessary, teaching the system about things it is missing improves recall, while teaching it about things it is getting wrong improves precision. This approach may entail taking a random sample of the documents suggested as responsive and checking for mistakes, or reviewing untrained documents to see where the system and reviewers disagree.

3. **Seed additional documents:** Using judgmental sampling to feed the system with items that are highly relevant can improve system learning and help improve recall.

4. **Verify training consistency:** Check your trained documents for inconsistency—inconsistent training can stifle machine learning and produce inconsistent results.

5. **Train random sample documents:** In some situations, additional training on a small random sample of documents helps improve metrics.

4.5 Validating the Results and Producing Documents

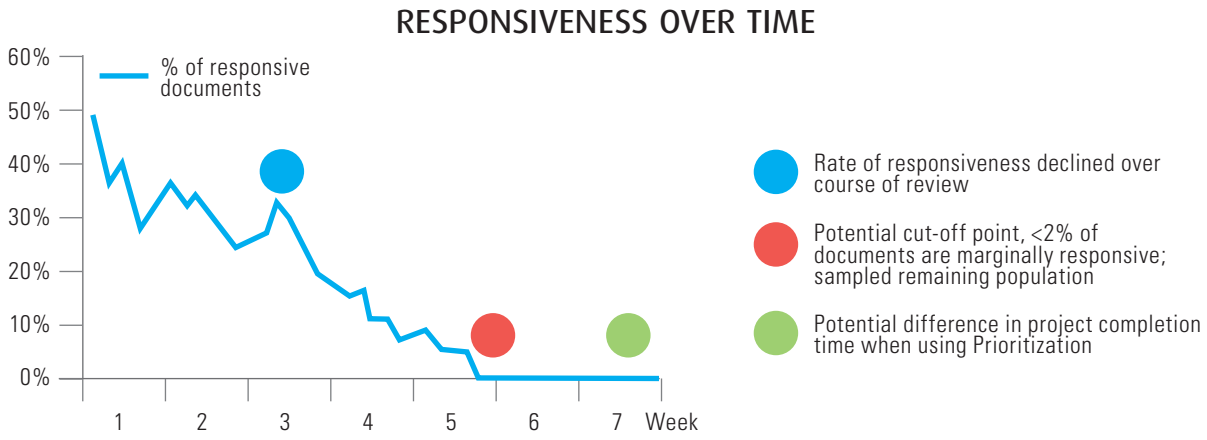
Deciding When to Stop Review

After sampling the data, the team may find that the machine is categorizing documents so effectively that manual review is no longer necessary. However, that means producing some documents without ever actually looking at them, which is likely a daunting thought for many teams concerned with defensibility. To ensure confidences and put those concerns to rest, sampling can help validate machine predictions, during which reviewers look at a fraction of the remaining documents to validate the machine's predictions. To illustrate two approaches used to end the predictive coding cycle, consider the case studies on the following pages.

Case Study #1 *(stop when the likelihood of finding responsive documents dwindles)*

In a project with nearly 500,000 documents, the prioritization component was no longer escalating relevant documents, and the categorization component suggested no relevant documents were left, with over 180,000 documents remaining.

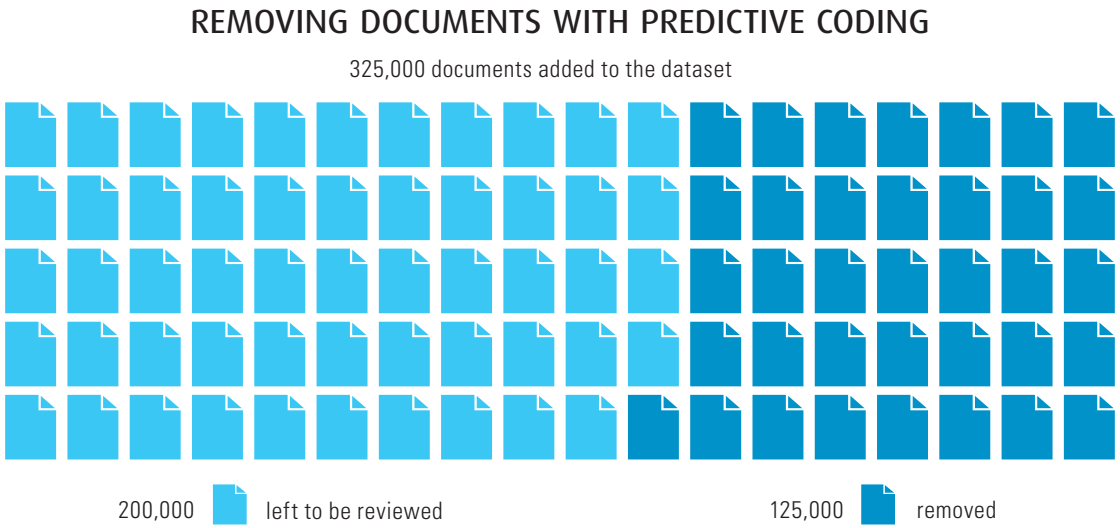
By conducting a statistical sample of approximately 9,200 documents, the team concluded that if more than 1% of the set was suggested responsive, they would continue review. The sample found only .5% of the set was suggested as responsive. Those documents were checked and found immaterial, meaning the team ended review with 37% of documents unreviewed by humans.



Case Study #2 *(team relies on the system to mass-categorize in lieu of manual review)*

In a matter with over 300 gigabytes of data and 750,000 documents, the client finds an additional 325,000 documents in the middle of review. With strict discovery deadlines and a limited budget, the team could not afford to perform manual review of all additional documents.

Trainers reviewed a 5% random sample of documents suggested as non-responsive with a high probability of correctness. The results showed a 94% agreement between trainers and categorization suggestions. The team relied on categorization suggestions to eliminate nearly 40% of the new documents from review—saving almost $200,000 in review costs.



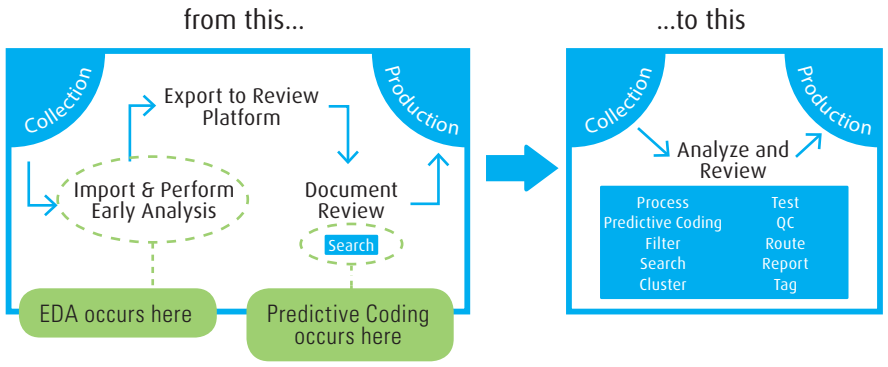
5. Predictive Coding Going Forward

Early commentary from the judiciary suggested that predictive coding was applicable in the right cases—namely cases with large data volumes. However, as familiarity with these advanced tools increases, predictive coding's value is extending beyond only large cases and the review process and further to the left on the EDRM. Simply put, the most pressing questions facing legal professionals today are not whether predictive coding should be used, but instead when should it be used as a part of your search methodology—and how can you leverage predictive coding for more than just document review.

Predictive Coding and Early Data Assessment

Early data assessment (EDA) aids fact-finding and narrows the data scope by helping attorneys understand their datasets by triaging data into critical and non-critical groups, identifying key players and critical case documents, and categorizing documents as efficiently as possible for review and production.

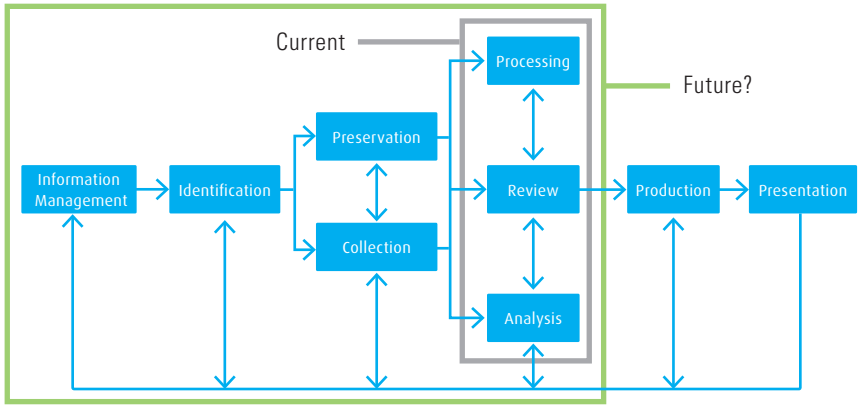
Although predictive coding and early data assessment are both keys to reducing costs and maximizing efficiency in ediscovery, they are often viewed as separate processes for litigation. In reality, these processes overlap substantially, and pointless inefficiencies are created by jockeying data between platforms and disparate processes. To reduce these inefficiencies, it is possible that an evolution will occur.



Predictive Coding for Information Governance

With the benefits of predictive coding now well-publicized and the number of use cases consistently growing, the most ardent supporters and progressive legal minds are moving toward the left side of the EDRM and considering predictive coding's use in the context of litigation or investigation readiness. Simply put, predictive coding is no longer restricted to reactive uses tied to a specific event. Instead, ediscovery professionals see its potential as a proactive, information governance tool.

ELECTRONIC DISCOVERY REFERENCE MODEL



2014 BEST OF THE NATIONAL LAW JOURNAL

Ediscovery.com Review, Kroll Ontrack's document review software, is a dynamic solution to conduct early data assessment, analysis, review and document production within a single platform. In 2014, Kroll Ontrack was named "Best Predictive Coding Ediscovery Solution" in The National Law Journal's annual reader's survey.

ediscovery.com Review

Kroll Ontrack provides technology-driven services and software to help legal, corporate and government entities as well as consumers manage, recover, search, analyze, and produce data efficiently and cost-effectively.

Kroll Ontrack®