insolvency 2020

Oct. 14, 2020, 1:30-3:00 p.m.

American Bar Association: Artificial Intelligence: Friend or Foe?

Warren E. Agin; Analytic Law, LLC
George Sanderson; Sanderson Law Firm PLLC
Kristi R. Smith; Bryan Cave Leighton Paisner LLP
Nellwyn W. Voorhies; Donlin Recano & Company, Inc.

Educational Materials

TECHSHOW2020

Hitchhiker's Guide to AI

WRITTEN BY:

Warren E. Agin

PRESENTERS:
Warren E. Agin: @AnalyticLaw
Thomas Hamilton: @TJHammy

December 23, 2019

A SIMPLE GUIDE TO MACHINE LEARNING

By Warren E. Agin[1]

Introduction

Lawyers know a lot about a wide range of subjects--the result of constantly dealing with a broad variety of factual situations. Nevertheless, most lawyers might not know much about machine learning and how it impacts lawyers in particular. In this article, I provide a short and simple guide to machine learning at a level understandable to the typical attorney.

The phrase "artificial intelligence" usually refers to machine learning in one form or another. It might appear as the stuff of science fiction, or perhaps academia, but in reality machine learning techniques are in broad use today. Such techniques recommend books for you on Amazon, help sort your mail, find information for you on Google, and allow Siri to answer your questions.

In the legal field, products built on machine learning are already starting to appear. Lexis and Westlaw now incorporate machine learning in their natural-language search and other features. ROSS Intelligence is an "AI" research tool that finds relevant "phrases" from within cases and other sources in response to a plain-language search. Through the use of natural language processing, you can ask ROSS questions in fully formed sentences and immediately receive highly relevant answers directly from primary law in a way that no other legal research tools can. Elevate Services' ContraxSuite uses machine learning to quickly analyze large numbers of contracts. These are just two of dozens of new, machine-learning-based products. On the surface, these tools might seem similar to current legal products, but you will see by the end of this article that they do something fundamentally different, making them not only potentially far more efficient and powerful, but disruptive as well. For example, machine learning is the "secret sauce" that enables ride-sharing services like Uber, allowing it to efficiently adjust pricing to maximize both the demand for rides and the availability of drivers, predict how long it will take a driver to pick you up, and calculate how long your ride will take. With machine learning, Uber and similar companies are rapidly displacing the traditional taxicab service. Understanding what machine learning is and what it can do is key to understanding its future effects on the legal industry.

What Is Machine Learning?

Humans are good at deductive reasoning. For example, if I told you that a bankruptcy claim for rent was limited to one year's rent, you would easily figure out the amount of the allowed claim. If the total rent claim were $100,000, but one year's rent was $70,000, you would apply the rule and deduce that the allowable claim is $70,000. No problem. You can determine the result easily, and you can also easily program a computer to consistently apply that rule to other situations. Now reverse the process. Assume I told you that your client was owed $100,000 and that the annual rent was $70,000, and then told you that the allowable claim was $70,000. Could you figure out how I got that answer? You might guess that

[1] Warren E. Agin is Managing Director of Digital Strategy and Solutions for Elevate Services, Inc. and Chair of the Business Law Section's Legal Analytics Committee. Mr. Agin thanks Michael Bommarito of Elevate Services and Thomas Hamilton of ROSS Intelligence for kindly reviewing and commenting on an earlier version of this article, but emphasizes that any errors are his, not theirs. This article is adopted from one previously published in the ABA Business Law Section's Business Law Today.

the rule is that the claim is limited to one year's rent, but could you be sure? Perhaps the rule was something entirely different. This is inductive reasoning, and it is much more difficult to do.

Machine learning techniques are computational methods for figuring out "the rules," or at least approximations of the rules, given the factual inputs and the results. Those rules can then be applied to new sets of factual inputs to deduce results in new cases.

Here is an example that is easy to understand. You all know the old number series games. For example:

2 4 6 8 10 _?_

The next number is 12, right? Here, the inputs are the series of numbers 2 through 10, and from this we induce the rule for getting the next number—add 2 to the last number in the series. Here is another one:

1 1 2 3 5 _?_

The next number is 8. This is a Fibonacci sequence, and the rule is that you add together the last two numbers in the series.

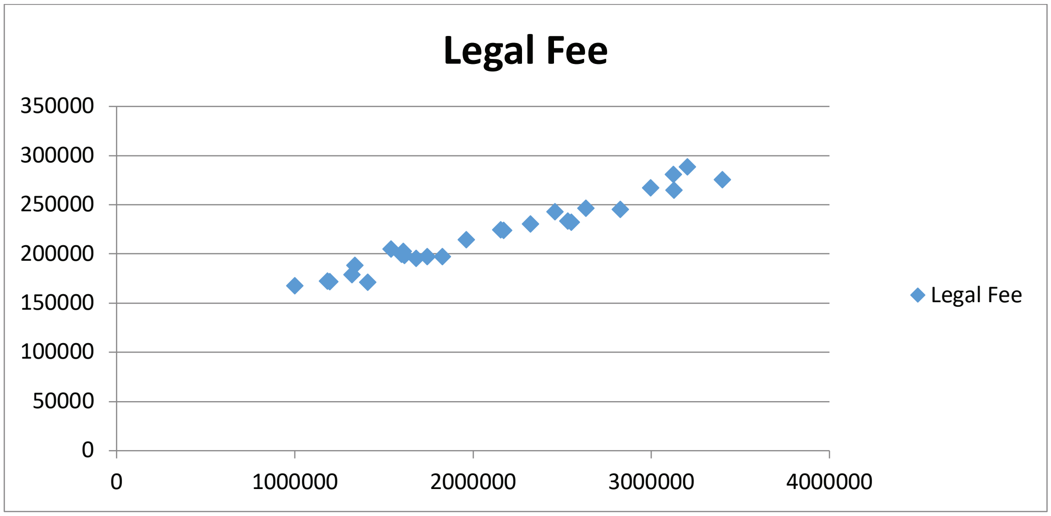
With these games what you are doing in your head is looking at a series of inputs and answers, and using inductive reasoning to figure out the rule. You then apply that rule to get the next number. Broken down a little, the prior game looks like this:

Input	Result
1 1	2
1 1 2	3
1 1 2 3	5
1 1 2 3 5	_?_

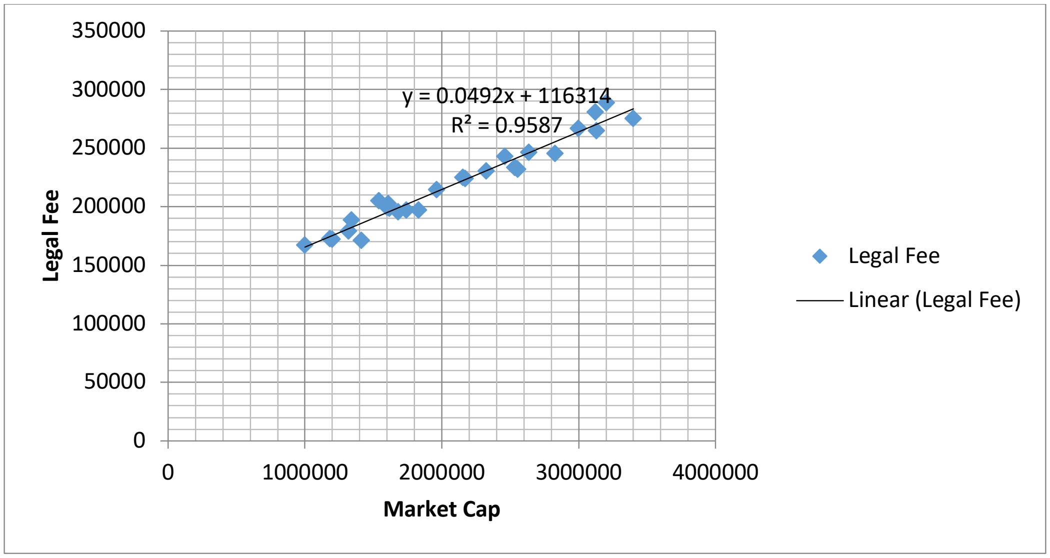
We look at the group of inputs and induce a rule that gives us the shown results. Once we have derived a workable rule, we can apply it to the last row to get the result "8," but more importantly we can apply it to any group of numbers in the Fibonacci sequence. This is a simple (very simple) example of what machine learning does.

Let's take a more complex example.

Assume we wanted to predict the amount of a debtor's counsel's fees in a Chapter 11 case. We could take a look at cases in the past and get information about each case; such as the number of creditors, the debtor's market capitalization, where the case was filed, and, of course, the eventual fee awarded to debtor's counsel. We might compare these numbers and discover that if we graphed the fee awards against the debtor's market capitalization, it looks something like this (purely hypothetically):



There seems to be a trend here. The larger the company market capitalization (the X-axis), the higher the legal fee seems to be. In fact, the data points look sort of like a line. We can calculate the line that best fits the data points using a technique called linear regression.



We can even see the equation that the line represents. You take the market capitalization for the debtor, multiply it by 4.92% and add $116,314. This is called a "prediction model." The prediction model might not perfectly fit the data used to create it – after all, not all the data points fall exactly on the line – but it provides a useful approximation. That approximation will provide a pretty good estimate for legal fees in future cases (that's what the R2 number on the graph tells us.) For the record, the data here is imaginary; hand tailored to demonstrate the methodology.

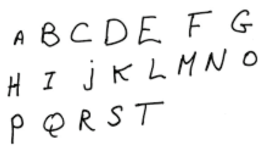
Naturally, real-world problems are more complex. Instead of a short series of numbers as inputs, a real-world problem might use dozens, perhaps thousands, of possible inputs that might be applied to an undiscovered rule to obtain a known answer. We also do not necessarily know which of the inputs are the ones our unknown rule uses!

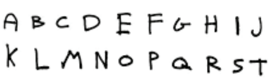
To solve a more complex problem, we might begin by building a database with the relevant points of information about a large number of cases, in each instance collecting the data points that we think might affect the answer. To build our prediction model, we would select cases at random to use as a "training set," putting the remainder aside to use as a "test set." Then we would begin to analyze the various relationships among the data points in our training set using statistical methods. Statistical analytics can help us identify the factors that seem to correlate with the known results and the factors that clearly do not matter.

Advanced statistical methods might help us sort through the various relationships and find an equation that takes some of the inputs and provides an estimated result that is pretty close to the actual results. Assuming we find such an equation, we then try it out on the test set to see if it does a good job there as well—predicting results that are close to the real results. If our predictive model works on our test set, then we consider ourselves lucky. We can now predict debtor's counsel's legal fees ahead of time; at least until changing circumstances - perhaps rules changes, a policy change at the US Trustee's Office, or the effect our very own model has on which counsel get hired for cases – renders our model inaccurate. If our model does not work on the test set data, than we consider it flawed and go back to the drawing board.

For real-world problems, this kind of analysis is difficult. The job of collecting the data, cleaning it, and analyzing it for relationships takes a lot of time. Given the large number of potential variables that affect real-world relationships, identifying those that matter is somewhat a process of trial and error. We might get lucky and generate results quickly, we might invest substantial resources without finding an answer at all, or the relationships might simply prove to be too complex for the methods I described to work adequately. Inductive reasoning is difficult to do manually. This brings us to machine learning. Machine learning can efficiently find relationships using inductive reasoning.

As an example of what machine learning can do, consider these images:





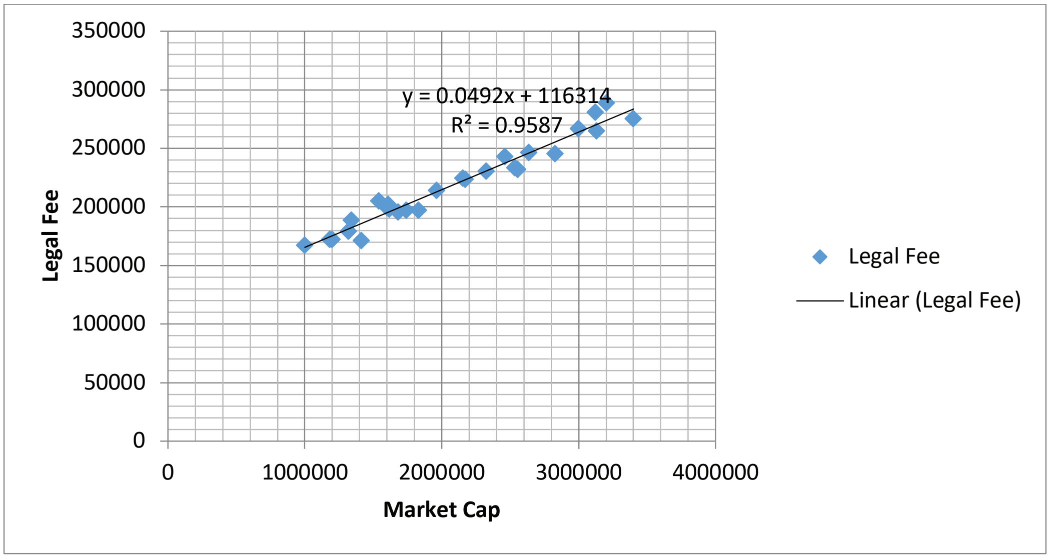
Assume we want to set up a computer system to identify these handwritten images and tell us what letter each image represents. Defining a rule set is too difficult for us to do by hand and come up with anything that is remotely usable, but we know there *is* a rule set. The letter A is clearly different from the letter P, and C is different from G, but how do you describe those differences in a way a computer can use to consistently determine which image represents which letter?

The answer is that you don't. Instead, you reduce each image to a set of data points, tell the computer what the image is of, and let the computer induce the rule set that reliably matches all the sets of data points to the correct answers. For the image recognition problem, you might begin by defining each letter as a 20 pixel by 20 pixel image, with each pixel having a different grey-scale score. That gives you 400 data points, each with a different value depending on how dark that pixel is. Each of these sets of 400 data points is associated with the answer--the letter they represent. These sets become the "training set," and another database of data points and answers is the "test set." We then feed that training set into our machine-learning algorithm—called a "learner"—and let it go to work.

What does the "learner" actually do? This is a little more difficult to explain, partially because there are a lot of different types of learners using a variety of methods. Computer scientists have developed a number of different kinds of techniques that allow a computer program to infer rule sets from defined sets of inputs and known answers. Some are conceptually easier to understand than others. In this article, I describe, in simple terms, how a couple of these techniques work. Machine learning programs will use a variation of one or more of these techniques. The most advanced systems include several techniques, using the one that fits the specific problem best or seems to generate the most accurate answers.

In general, think of a learner as including four components. First, you have the input information from the training set. This might be data from a structured, or highly defined, database, or unstructured data like you might find in a set of discovery documents or in a collection of websites. Second, you have the answers. With a structured database, a particular answer will be closely identified with the input information. With unstructured information, the answer might be a category, such as which letter an image represents or whether a particular e-mail is spam; or the answer might be part of a relationship, such as text in a court decision that relates to a legal question asked by a researcher. Third, you have the learning algorithm itself—the software code that explores the relationships between the input information and the answers. Finally, you have weighting mechanisms—basically parts of the algorithm that help define the relationships between the input information and the answers, within the confines of the algorithm. Once you have these four components, the learner simply adjusts the weighting mechanisms in a controlled manner until it finds values for the weighting mechanisms that allow the algorithm to accurately match the input information with the known correct answers.

Let's see how this might work with my hypothetical system for estimating debtor's counsel's fees. Here, for reference, is the graph again.



In the example, the market capitalizations are the input information ("X"). The known legal fees for each case are the answer ("Y"). For purposes of illustration, let's assume the algorithm is Y=aX+b (a vast simplification, but I'm going to use it to demonstrate a point). The weighting mechanisms are the two variables "a" and "b." Instead of manually calculating the values of "a" and "b" using linear regression, a machine learning program might instead try different values of "a" and "b," each time checking to see how well the line fits the actual data points mathematically. If a change in "a" or "b" improves the fit of the line, the learner might continue to change "a" and "b" in the same direction, until the changes no longer improve the line's fit.

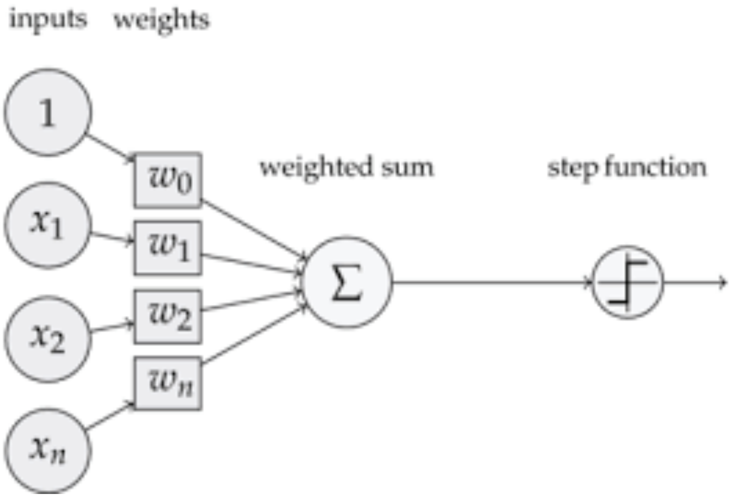
Of course, in my example it is easier just to calculate "a" and "b" using linear regression techniques. I don't even need to have math skills to do it – the functionality is built right into Microsoft Excel and other common software products. Given a spreadsheet with the data, I can perform the calculation with a few mouse-clicks. Machine learning programs, however, can figure out the relationships when there are millions of data points and billions of relationships—when modeling the systems is impossible to do by hand because of the complexity. Machine learning systems are limited only by the quality of the data and the power of the computers running them.

Now, let's look at a couple of types of machine learning systems.

Neural Networks

The term "neural network" conveys the impression of something obscure and mysterious, but it is probably the easiest form of a machine learning system to explain to the uninitiated. This is because it is made up of layers of a relatively simple construct called a "perceptron."

Meet a perceptron.



Credit: https://blog.dbrgn.ch/2013/3/26/perceptrons-in-python/

This perceptron contains four components, the first being one or more inputs represented by the circles on the left. The input is simply a number, perhaps between 0 and 1. It might represent part of our input information, or it might be the output from another perceptron.

Second, each input number is given a weight—a percentage by which the input is multiplied. For example, if the perceptron has four inputs of equal importance, each input is multiplied by 25 percent. Alternatively, one input might be multiplied by 70 percent while the other three are each multiplied by 10 percent, reflecting that one input is far more important than the others.

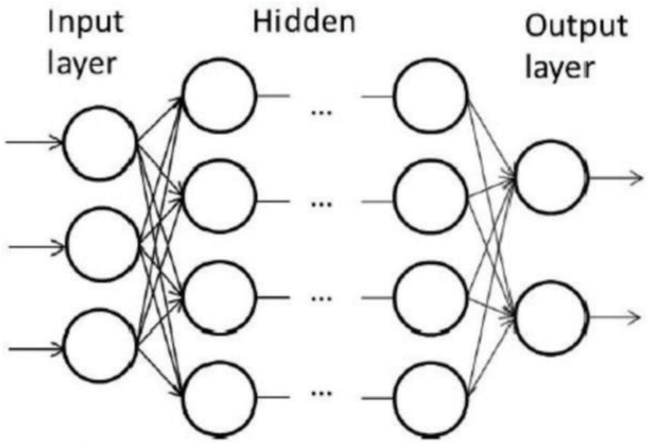
Third, these weighted input numbers are added to generate a weighted sum—a single number that reflects the weights given the various inputs.

Fourth, the weighted sum is fed into a step function. This is a function that outputs a single number based on the weighted sum. A simple step function might output a "0" if the weighted sum is between 0 and .5, and a "1" if the weighted sum is between .5 and 1. Usually a perceptron will use a logarithmic step function designed to generate a number between, say, 0 and 1 along a logarithmic scale so that most weighted values will generate a result at or near 0, or at or near 1, but some will generate a result in the middle.

Some systems will include a fifth element: a "bias." The bias is a variable that is added or subtracted from the weighted sum to bias the perceptron toward outputting a higher or lower result.

In summary, the perceptron is a simple mathematical construct that takes in a bunch of numbers and outputs a single number. That output number might be fed to another perceptron, or it might relate to a particular "answer." For example, if your learner is doing handwriting recognition, you might have a perceptron that tells you the image is the letter "A" based on whether the output number is closer to a 1 than a 0.

In a neural network, the perceptrons typically are stacked in layers. The first layers receive the input information for the learner, and the last layer outputs the results.



(a) Arquitecture of multilayer perceptron

Credit: http://www.intechopen.com/books/cerebral-palsy-challenges-for-the-future/brain-computer-interfaces-for-cerebral-palsy

In between are what are called "hidden layers" of perceptrons, each taking in one or more input numbers from a prior layer and outputting a single number to one or more perceptrons in the next layer.

The computer scientist building the neural network determines its design--how many perceptrons the system uses, where the input data comes from, how the perceptrons connect, what step function gets used, and how the system interprets the output numbers. However, the learner itself decides what weights are given to each input as the numbers move through the network, and what biases are applied to each perceptron. As the weights and biases change, the outputs will change. The learner's goal is to keep adjusting the weights and biases used by the system until the system produces answers using the input information that most closely approximates the actual, known answers.

Returning to the handwriting recognition example, remember that we broke down each letter image into 400 pixels, each with a greyscale value. Each of those 400 data points would become a input number into our system and be fed into one or more of the perceptrons in the first input layer. We add some hidden layers in the middle. Finally, we would have an output layer of 26 perceptrons, one for each letter. The output perceptron with the highest output value will tell us what letter the system thinks the image represents.

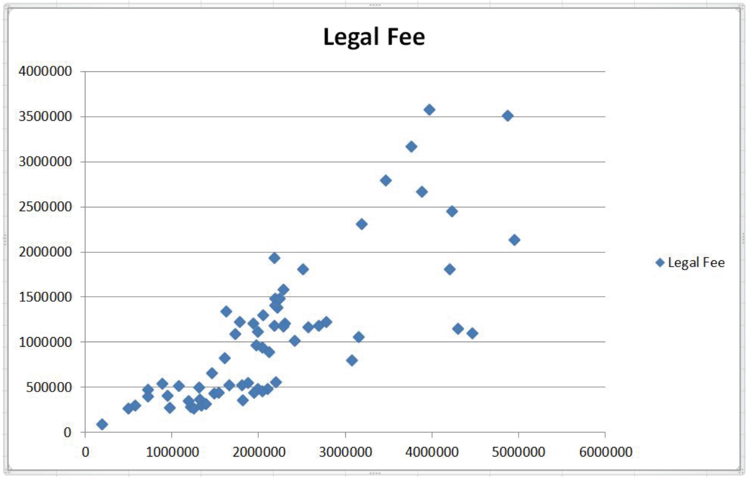
Then, we pick some initial values for the weights and biases, run all the samples in our training set through the system, and see what happens. Do the output answers match the real answers? Probably not even close the first time through. So, the system begins adjusting weights and biases, with small, incremental changes, testing against the training set and continuously looking for improvements in the results until it becomes as accurate as it is going to get. Then, the test set is fed into the system to see if the determined set of ideal weights and biases produces accurate results. If it does, we now have an algorithm that we can use to interpret handwriting.

It might seem a little like magic, but even a relatively simple neural network, properly constructed, can be used to read handwriting with a high degree of accuracy. Neural networks are particularly good at sorting things into categories, especially when using a discrete set of input data points. What letter is it? Is it a picture of a face or something else? Is a proof of claim filed in a bankruptcy case objectionable or not?

Nearest Neighbor

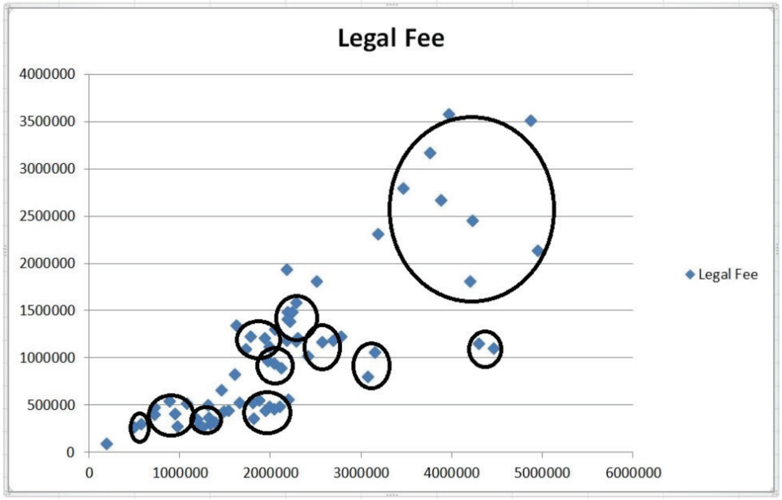
The k-nearest neighbor or k-NN algorithm makes a good second choice because its name at least makes it sound easy to understand. But, it is, in fact, one of the simplest of machine learning algorithms. k-NN algorithms group items into categories based on similarity of characteristics. We can analogize k-NN to the way that we humans actually think about qualitative objects. For example, we hold a reference set of accessible, salient memories (e.g., fruits). These memories are encoded along a certain set of dimensions (e.g., color, size, shape). When you are asked to categorize them or recall a similar fruit, you're essentially carrying out k-NN (though each of our encodings and reference sets may vary based on experience).

To explain how it works, let's go back to a variation of our examination of debtor's counsel's fees. Instead of that nice linear relationship we saw before, let's assume the data looks more like this:

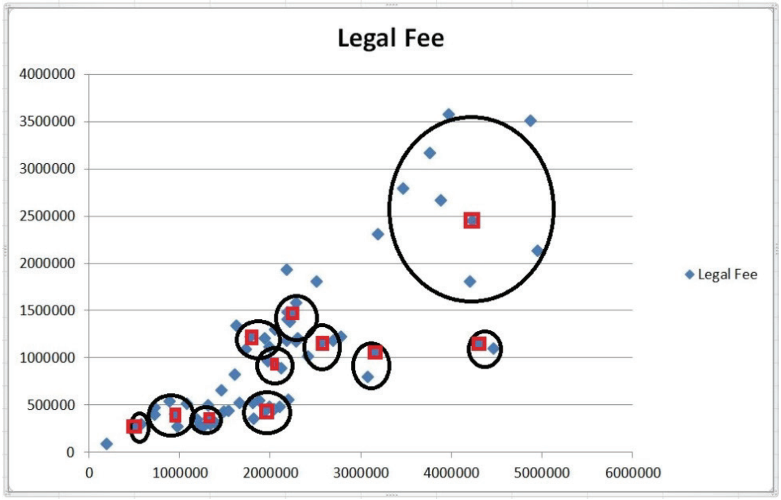


Again, the Y axis (on the bottom) is the debtor's market capitalization, and the X axis (on the left) is debtor's counsel's legal fee. This relationship is still pretty linear, but with a lot of variation, especially for the larger companies.

k-NN would start by measuring the linear distance between the data points. Each data point would be compared with the other data points near it, using the "k" variable as a threshold to determine how far out the algorithm will look. The algorithm will discover that some dots are surrounded by lots of close neighbors. Others are on the outside of a grouping, with a few close neighbors. By seeing how far away and in which direction a dot's neighbors are, the algorithm can start to determine which dots are similar to each other, and can start to group them. A particular dot will be assigned to the group of dots to which it has the closest relationships.



Once the algorithm reaches this point, it can start to reduce the data set to a reduced set of reference points, each of which represents a group. These reference points will be the dot or data point located closest to the mean value for the grouping.



The learner has now completed its work, building a set of known reference points that each represent a different category of potential data within the entire data set.

This particular learner doesn't help us predict the legal fees in a case given the company's market capitalization, as much as it does something equally interesting – it helps us categorize cases given both the market capitalization and the amount of debtor's counsel's legal fees. When the next case comes around, we can categorize it and group it with similar cases, based on the spatial distance between the new case's data point and the various reference points. We assign the new case to the grouping or category represented by its nearest neighbor. We might not know why the case is the same as the others, but we can identify the existence of a relationship.

This might be very helpful. For example, we might note that the cases in a particular group always end up back in a second Chapter 11 case. Now, even without having any particular knowledge of why this might be, we might conclude that future cases assigned to this group will also end up in a second Chapter 11 case.

In real life situations, the learner isn't working with just two dimensions – or data inputs. It might be working with dozens of characteristics for each data point. Machine learning takes place in a multi-dimensional environment, which makes it very hard to visualize, even if the math doesn't change all that much.

k-NN and algorithms like it play an important role in interpreting unstructured data and tasks like natural-language processing. They can identify relationships among words or concepts. The computer does not understand the words, what the concepts are, or what they mean, but it can identify the relationships as relationships and, like a parrot that repeats what it hears, convey the impression of understanding.

Machine Learning in Action

My examples are basic, designed to provide some understanding of what are fairly abstract systems. Machine learners come in many flavors—some suitable for performing basic sorting mechanisms, and others capable of identifying and indexing complex relationships among information in unstructured databases. Some systems work using fairly simple programs and can run on a typical office computer, and others are highly complex and require supercomputers or large server farms to accomplish their tasks.

To understand the power of machine learning systems compared with non-learning analytic tools, let's revisit an earlier example in the article: ROSS Intelligence. While ROSS was originally partially built on the IBM Watson system, over the years it has developed its own completely proprietary machine learning techniques to perform its tasks. These search tools employ a number of machine learning algorithms working together to categorize semantic relationships in unstructured textual databases. In other words, if you start with a large database of textual material dealing with a particular subject, machine learning tools can begin by indexing the material, noting the vocabulary and which words tend to associate with other words. Even though these systems do not actually understand the text's meaning, they develop, through this analysis, the ability to mimic understanding by finding the patterns in the text.

For example, when you conduct a Boolean search in a traditional service for "definition /s 'adequate protection,'" the service searches its database for an exact match for those terms applying the Boolean search logic provided. ROSS does something different. It looks within the search query for word groups it recognizes and then finds the results it has learned to associate with those word groups. If you search for "what is the definition of adequate protection" the system will associate the query "what is the definition" with similar queries, such as "what is the meaning of" or just "what is." It will also recognize the term "adequate protection" as a single concept instead of two separate words, and likely, given the context, understand it as a word found in bankruptcy materials. Finally, it will have associated a successful response as being one that gives you certain types of clauses including the term "adequate protection." It won't understand specifically that you are looking for a definition, but because others who used the system and made similar inquiries preferred responses providing definitions, you will get clauses containing similar language patterns and, viola, you will get your definition.

You should not even have to use the term "adequate protection" to get an answer back discussing the concept when that is the appropriate answer to your question. So long as your question triggers the right associations, the system will, over time, learn to return the correct responses.

The key is that a machine learning system learns. In a way, we do the same thing ROSS does. The first time we research a topic, we might look at a lot of cases and go down a lot of dead ends. The next time, we are more efficient. After dealing with a concept several times, we no longer need to do the research. We remember what the key case is, and at most we check to see if there is anything new. We know how the cases link together, so the new materials are easy to find.

A machine-learning-based research tool can do this on a much broader scale. It learns not just from our particular research efforts, but from those of everyone who uses the system. As the system receives more use, it continues to use user feedback to assess how its model performs and allow for periodic retraining. As a result, it will become extremely adept at providing immediate responses to the most common queries by users. It might also be able to eventually give you a confidence level in its answer, comparing the information it provides against the entire scope of reported decisions and its users' reactions to similar, prior responses, to let you know how reliable the results provided might be. Even though the system doesn't understand the material in the same manner as a human, its ability to track relationship building over a large scope of content and a large number of interactions allow it to behave as you might, if you had researched a particular point or issue thoroughly many times previously. This provides a research tool far more powerful than existing methodologies.

Legal tools based on machine learning have enormous application. Learners already in use by lawyers help with legal research, categorize document sets for discovery purposes, evaluate pleadings and transactional documents for structural errors or ambiguity, perform large-scale document review in M&A, or identify contracts affected by systemic change—like Brexit or the LIBOR crisis. General Motors' legal department along with other large companies are exploring using machine learning techniques to evaluate and predict litigation outcomes and even help choose which law firms they employ. Machine learning is not the solution for every question, but it can help answer a large number of questions that simply were not answerable in the past, and that is why the advent of machine learning in the legal profession will prove truly transformational.

Georgia State University Law Review

Volume 35
Issue 4 *Summer 2019*

Article 2

6-1-2019

Predicting Chapter 11 Bankruptcy Case Outcomes Using the Federal Judicial Center IDB and Ensemble Artificial Intelligence

Warren E. Agin
Elevate Services, Inc., LexPredict, agin@analyticlaw.com

Gill Eapen
Decision Options, LLC, gill@eapen.info

PREDICTING CHAPTER 11 BANKRUPTCY CASE OUTCOMES USING THE FEDERAL JUDICIAL CENTER IDB AND ENSEMBLE ARTIFICIAL INTELLIGENCE

Warren E. Agin* & Gill Eapen**

INTRODUCTION

Over 100,000 Chapter 11 bankruptcy cases were filed in the United States over the ten-year period from 2008–2017.[1] These cases represent a cross section of society; from large, public corporations to small mom-and-pop stores to individuals trying to work out real estate investments.[2] Regardless of whether the filer was a corporation or an individual, a large entity or a small business, each case shared a common goal—to use the provisions of Chapter 11 of the United States Bankruptcy Code to reorganize its assets, operations, and financial affairs and hopefully return to profitability.[3] Despite large investments in judicial resources, more than half of these cases failed to achieve their goals.[4]

Understanding how a case is likely to end has very real value for practitioners. Chapter 11 bankruptcy cases almost always resolve in

* Senior Consultant, Elevate Services, Inc., LexPredict.

** CEO, Decision Options, LLC. The authors want to thank Jonathan Boyarsky, Boston College School of Law class of 2021, for his assistance in preparing this paper.

1. This information was calculated using data obtained from the Federal Judicial Center, downloaded from its Integrated Database at https://www.fjc.gov/research/idb/interactive/IDB-bankruptcy. *See also Statistics From Epiq Systems*, AM. BANKR. INST., https://www.abi.org/newsroom/bankruptcy-statistics [https://perma.cc/WF6E-UYXT] (last visited Feb. 22, 2019).
2. Edith H. Jones, *Chapter 11: A Death Penalty for Debtor and Creditor Interests*, 77 CORNELL L. REV. 1088, 1088–89 (1992).
3. *Id.*
4. *Id.* At one point, only 10% of Chapter 11 cases resulted in confirmed plans of reorganization. *Id.* More recent studies show a confirmation rate between 30% and 33%, depending on the sample. Elizabeth Warren & Jay L. Westbrook, *The Success of Chapter 11: A Challenge to the Critics*, 107 MICH. L. REV. 603, 615 (2009). Our analysis of ten years of case data taken from the IDB suggests a confirmation rate of up to 45%, although this statistic is qualified by the caveat that confirmation of a Chapter 11 plan must be inferred when using the IDB. S*ee Statistics From Epiq Systems*, *supra* note 1.

one of three ways. Some cases end up dismissed by the court.[5] Others end up converted into Chapter 7 cases, where operations cease, and assets are liquidated for distribution to creditors.[6] The remaining cases achieve some level of success: either obtaining confirmation of a plan of reorganization or managing to operate under the protection of Chapter 11 long enough to liquidate in an orderly fashion as an operating entity.[7] The attorneys, financial advisors, credit managers, and others involved in Chapter 11 cases often face difficult decisions about how to respond to these businesses in crisis.

In this project, the authors obtained public data on over 100,000 Chapter 11 bankruptcy cases and used machine and deep-learning methodologies to explore whether models could be designed to predict Chapter 11 case outcomes. The data used was obtained from the Federal Judicial Center's bankruptcy Integrated Database and included information about case filing dates, the court where the case was filed, the type of business entity, and basic information about assets and liabilities. Using this information, the authors initially sought to predict whether a particular case was dismissed, converted to another Chapter under the Bankruptcy Code, or closed with a plan. Cases that had not yet closed at the end of the dataset's period, but had been open in Chapter 11 for at least two years, were treated as viable Chapter 11 cases. Of the cases used for the project, about 55% were dismissed or converted. The authors also created models to predict between two outcomes—dismissal or conversion, as opposed to case viability.

The authors used most commonly known and reasonably applicable machine learning algorithms and deep-learning optimizers in an attempt to create a robust model. The general-purpose AI platform, Decision Options®, was utilized to explore the data and build meta-models. We achieved an accuracy of about 75% for the selected models. These results show the ability of AI systems to

5. *See* 11 U.S.C. § 1112 (2018) (governing conversion and dismissal of Chapter 11 bankruptcy cases).
6. *See* 11 U.S.C. §§ 701–84 (2018).
7. Warren & Westbrook, *supra* note 4, at 615.

predict Chapter 11 case outcomes with some level of accuracy with even limited data, exceeding baseline statistics. However, the project also highlighted the deficiencies in the data made publicly available in formats that machine learning systems can easily use and the promise of significantly better results with higher quality information.

I. Chapter 11 Bankruptcy and Potential Case Outcomes

The Chapter 11 bankruptcy process provides a mechanism for businesses and individuals to restructure their debts and other obligations using a flexible mechanism.[8] Although it is used primarily by business organizations, individuals too can seek protection from creditors using Chapter 11.[9] The primary mechanism used to reorganize is the Chapter 11 plan of reorganization. A Chapter 11 plan allows a company to restructure its debt and capital, shed undesired business arrangements and contracts, cure outstanding defaults on valuable contracts, and provide buyers of assets with protection against prior creditors.[10] Sometimes Chapter 11 cases are successful, resulting in a reorganization of business assets and debts to increase stakeholder values. Sometimes, the cases are not successful. Can the results of a bankruptcy case be predicted in advance? A small number of prior studies have addressed this question, using data on a limited number of large bankruptcy cases.

Once a Chapter 11 case commences, four possible outcomes exist. First, the Chapter 11 case may end up dismissed,[11] removing the debtor from the bankruptcy court's protection. Second, the case may be converted to a case under a different Chapter of the Bankruptcy Code—typically Chapter 7, where a trustee is appointed to liquidate the debtor's assets.[12] Both of these outcomes can generally be

8. *See* 11 U.S.C. §§ 1101–16 (2018).
9. 11 U.S.C. § 101(35) (2018) (defining the term "person"); 11 U.S.C. § 109.
10. Jones, *supra* note 2, at 1089.
11. 11 U.S.C. § 1112(b) (2018).
12. 11 U.S.C. § 1112 (2018). Generally, a business debtor cannot operate during a Chapter 7 case. *Bankruptcy: What Happens When Public Companies Go Bankrupt*, U.S. SEC. & EXCHANGE

considered a failure of the reorganization process. Third, a debtor's assets may be sold, either through confirmation of a Chapter 11 plan or through a successful sale process.[13] A sale does not necessarily evidence a failure of the Chapter 11 process; in many cases, a single buyer will purchase substantially all of the debtor's assets through an acquisition entity which then continues to operate the business. To the public, the change in ownership may be completely invisible. Fourth, the Chapter 11 plan process helps to restructure the debtor's finances, allowing the debtor to continue business operations after Chapter 11 or change ownership of the business through a merger or acquisition, with the debtor's business continuing to operate as a separate entity.

In theory, the company that can successfully restructure its finances and operations and stay in business will generate more value for its stakeholders than might be obtained from a straightforward sale of assets.[14] Although many cases result in an asset sale, assets sold out of an operating business will generally obtain better prices than assets sold out of a closed business, and an orderly liquidation obtains better results than a disorderly scramble by creditors. As a result, understanding how a particular bankruptcy case may resolve has value to everyone involved in the process.

Prior studies attempting to predict Chapter 11 outcomes have primarily sorted cases into those where the company continued as an

---

COMMISSION (Feb. 3, 2009) https://www.sec.gov/reportspubs/investor-publications/investorpubsbankrupthtm.html [https://perma.cc/D697-W3KK]. Typically, business operations end and any remaining assets are simply liquidated. *Id.*

13. 11 U.S.C. § 363 (2018). This sale process is referred to as a "363 Sale" and named after the section of the Bankruptcy Code used to authorize asset sales without using the Chapter 11 plan-confirmation process. *What Is a 363 Sale?*, CORP. FIN. INST., https://corporatefinanceinstitute.com/resources/knowledge/deals/363-sale/ [https://perma.cc/3G3P-YXWG] (last visited Feb. 20, 2019). In larger Chapter 11 cases, the § 363 sale is typically followed by confirmation of a plan addressing disposition of funds and handling of remaining case issues. *Id.* For smaller cases, a 363 sale might be followed by conversion to Chapter 7 to allow the Chapter 7 trustee to distribute the sale proceeds. *Id.*

14. *See* Michael Bradley & Michael Rosenzweig, *The Untenable Case for Chapter 11*, 101 YALE L.J. 1043, 1095 (1992); Robert K. Rasmussen & Douglas G. Baird, *Chapter 11 at Twilight* 1–24 (John M. Olin Program in Law & Econ. Working Paper No. 201, 2003).

operating entity post-bankruptcy and those where assets were sold.[15] This dichotomy may be dictated by the nature of the cases for which data was available: large corporations and primarily public companies. Large public-company cases rarely end up dismissed, and they rarely end up converted to a case under Chapter 7. When a large, public-company case is not successful, the result is usually an accelerated sale of assets through a Chapter 11 plan, or a series of asset sales using § 363 followed by a Chapter 11 plan that controls the remainder of the company's winding-down process.

The most useful data set for examining large-company Chapter 11 outcomes was assembled by Professor Lynn LoPucki and the UCLA School of Law. Known as the Bankruptcy Research Database (BRD), it contains information on a little over 1,000 large public company cases. In 2015, using the BRD, Lynn LoPucki and Joseph Doherty published *Bankruptcy Survival*, which sought to build a regression-based prediction model for evaluating whether a particular case would result in a continuing operating business or liquidation.[16] The study examined a subset of 634 cases from the BRD, of which 70% were classified as surviving.[17]

LoPucki and Doherty ran a series of logistic regressions[18] on about seventy variables tracked within the BRD using survival as the dependent variable.[19] Subsets of these seventy variables were tested in hundreds of combinations to ascertain the combination of variables that best correlated with corporate survival in the Chapter 11

15. Lynn LoPucki & Joseph Doherty, *Bankruptcy Survival*, 62 UCLA L. REV. 970, 979 (2015); Jairaj Gupta & Mariachiara Barzotto, Insights on Bankruptcy Emergence (Aug. 19, 2018) (unpublished manuscript) (on file at https://ssrn.com/abstract=3216433 [https://perma.cc/P44M-BT4Q]).

16. *Id.* at 978.

17. *Id.* at 983.

18. *Id.* at 978. Logistic regression is a form of regression analysis designed to predict categorical choices, such as, in this instance, whether a particular company "survived" Chapter 11. *Id.* For more on logistic regression, see *Lesson 6: Logistic Regression*, PA. STATE EBERLY COLL. OF SCI., https://onlinecourses.science.psu.edu/stat504/node/149/ [https://perma.cc/7X4A-9RAC] (last visited Feb. 22, 2019).

19. LoPucki & Doherty, *supra* note 15, at 979. In most prediction models, a series of data points referred to as "independent variables" or "features" are used to predict the "dependent variable" or "label." *Id.* In other terms, the independent variables are the things that you know, and the dependent variable is the thing that you want to know. *See id.*

process.[20] They identified three different models that did equivalent jobs of predicting case survival. All three models contained ten independent variables:

- Whether the company provided advance notice of an intention to sell its assets;
- Whether the company's EBIT[21] was positive;
- The shareholder to equity ratio;[22]
- Whether the company was in the manufacturing sector;
- The prime rate of interest one year prior to the petition date;
- The distance between the company's headquarters and its local bankruptcy court;[23]
- Whether a plan was pre-negotiated;
- Whether a debtor-in-possession loan was obtained;[24]
- Whether a creditors' committee was appointed; and
- The size of the company in asset value.[25]

In addition to the ten variables used in all three models, one model added the log value of the judge's years of experience, one added whether the case was filed in Delaware or New York as opposed to

20. LoPucki & Doherty, *supra* note 15, at 981.
21. *Id.* at 1000. EBIT means Earnings Before Interest and Taxes, which is a measure of operational profitability. *Id.*
22. *Id.* at 1004.
23. LoPucki & Doherty, *supra* note 15, at 992. The log value of the distance was used. *Id.* at 986. The distance is calculated to the local bankruptcy court, not the court where the case was filed. *Id.* at 992. The paper suggests that the variable served as a substitute for whether the company was geographically isolated. *Id.*
24. *Id.* at 1001. A DIP, or debtor-in-possession, loan refers to specialized post-petition financing obtained to capitalize operations during the Chapter 11 process. *Debtor-In-Possession (DIP) Financing Can Help Turn a Company Around Following Bankruptcy*, PARAGON FIN. GROUP, https://www.paragonfinancial.net/how-factoring-works/articles-resources/factoring-articles/debtor-in-possession-dip-financing-company-bankruptcy/ [https://perma.cc/GYY2-FYBN] (last visited Jan. 23, 2019). A DIP loan requires court approval and typically provides greater protection for the lender than the typical non-bankruptcy lending transaction. *Id.*
25. LoPucki & Doherty, *supra* note 15, at 1008 (the log of the asset value was used).

another jurisdiction, and the third model added a variable indicating whether the judge had presided over at least six prior large cases.[26] Each model presented a pseudo-R-squared value of about .26,[27] but there was no information about each model's prediction accuracy.

The article *Insights on Bankruptcy Emergence*—proposing the use of a regression-based model to predict the likelihood of a company emerging from Chapter 11 successfully—built on the LoPucki-Doherty study and examined the correlation between a variety of variables and success in Chapter 11 cases.[28] The analysis was based on examining 401 Chapter 11 filings using data collected from the BRD, coupled with relevant financial data obtained from Compustat.[29] The created model used eight features to achieve a classification performance of 94%.[30] Relevant factors identified in this study include filing in a debtor-friendly jurisdiction, having a high asset-to-debt ratio, being outside the retail industrial business sector, replacing the CEO after filing, having a pre-negotiated or pre-packaged plan, and a high debtor-in-possession loan-to-assets ratio.[31] This study, conducted by Jairaj Gupta and Mariachiara Barzotto, also found that filers are significantly less likely to emerge from Chapter 11 intact when they announce, at the start of the case, an intent to sell substantially all assets and when a significant amount of time passes before plan confirmation.[32]

26. *Id.* at 985, 990–91.

27. *Id.* at 986. The R-squared statistic is a measure of the relationship between the independent values used and the dependent variable. *Lesson 1.5: The Coefficient of Determination*, PA. STATE EBERLY COLL. SCI., https://onlinecourses.science.psu.edu/stat501/node/255/ [https://perma.cc/VL9K-USC3] (last visited Feb. 22, 2019). The closer the R-squared number is to one, the stronger the relationship. *Id.* An R-squared value of .26 can be interpreted as meaning that the eleven data points considered explain 26% of the variation in success outcomes. *Id.*

28. *Id.* at 2–3.

29. *Id.* at 3. Made available through Standard & Poor's, the Compustat database contains financial, statistical, and market information on active and inactive global companies throughout the world. *Fundamental Data*, S&P GLOBAL, https://www.spglobal.com/marketintelligence/en/solutions/fundamental-data [https://perma.cc/LUH6-A2DV] (last visited Apr. 9, 2019).

30. Gupta & Barzotto, *supra* note 15, at 4. The performance was measured using an AUC (area under the curve) metric. *Id.* The Gupta-Barzotto model also had a pseudo R-squared of .55, compared with the .26 metric for the LoPucki-Doherty model. *Id.*

31. *Id.*

32. *Id.* at 4–5, 14.

Georgia State University Law Review, Vol. 35, Iss. 4 [2019], Art. 2

Gupta and Barzotto's project built a prediction model to tell whether a company filing Chapter 11 would emerge successfully, defined as when the confirmed Chapter 11 plan either had the company continuing as an independent entity or being acquired through a merger or stock acquisition.[33] The project deemed a Chapter 11 case unsuccessful when the company's Chapter 11 case was dismissed or converted to Chapter 7, or the company's assets were sold in lieu of a merger.[34]

Gupta and Barzotto built their model using multivariate probit regression[35] applied to groups of potential features and then using the features showing the most predictive value.[36] The final model used eight features.[37] Although the classification accuracy of the model was an impressive 94%, the inclusion of one feature in particular calls into question its true predictive power.[38] The model included the length of time, measured in years, from the start of the case to case disposition.[39] This is not information available at the beginning of a Chapter 11 case and, thus, is inappropriate to include in a model attempting to predict, in advance, case results. Also, although not explicitly stated, the 94% accuracy statistic appears to derive from applying the model to the same data used to build the model. In machine-learning terms, the statistic is from the training set instead of a separate-test set.[40] Regression models, like any type of machine-learning model, can easily overfit to the data available. In other words, a model that is overfit to its data simply describes what is going on with the data used to create the model but fails to accurately predict results when applied to new cases. Because the model was built on 401 data samples, this draws into some question the model's

33. *Id.* at 4–5, 7.
34. Gupta & Barzotto, *supra* note 15, at 15.
35. Multivariate probit regression is a methodology similar to logistic regression, referenced *supra* note 18.
36. Gupta & Barzotto, *supra* note 15, at 21.
37. *Id.* at 22.
38. *Id.* at 31.
39. *Id.* at 7.
40. *Id.* at 31–32.

ability to predict results against new data. Even so, the authors' ability to reach such a high level of accuracy using a limited number of independent variables demonstrates the strong relationships between initial case information and case results, as well as how statistical systems are capable of describing those relationships.

In *What Drives Bankruptcy Forum Shopping? Evidence from Market Data*, Professor Jared Ellias examined factors behind predicting Chapter 11 case outcomes and concluded that insolvency professionals are better able to predict results for cases filed in Delaware and the Southern District of New York.[41] For his study, he examined data collected on 285 large, corporate bankruptcies combined with pricing information for related financial contracts.[42] He calculated the pricing deviation for each financial contract by taking the square of the gain or loss on the investment when purchased at the start of the bankruptcy case.[43] In theory, the lower the pricing deviation, the more accurately the investors were able to price the financial instrument early in the bankruptcy case. So, although Professor Ellias was not trying to predict outcomes, his methodology was designed to identify the case factors that allow for more accurate prediction.

The study used regression models to evaluate the factors that contribute to more accurate prediction of case outcomes.[44] It found a persistent and statistically significant relationship between filing in Delaware or the Southern District of New York and accuracy in predicting outcomes.[45] In other words, financial investors were more accurate predicting outcomes for cases filed in those two judicial districts. Although the paper focused on this aspect of predictability, other statistically significant factors increasing the predictability of a Chapter 11 case were the presence of private equity in the ownership

---

41. Jared A. Ellias, *What Drives Bankruptcy Forum Shopping? Evidence from Market Data*, 47 J. LEGAL STUD. 119, 119 (2018).
42. *Id.* at 121.
43. *Id.*
44. *Id.* at 146.
45. *Id.* at 122.

structure and the existence of a prepackaged Chapter 11 plan (although pre-negotiated plans did not have the same effect).[46]

These recent studies using the BRD and other smaller datasets apply sophisticated regression techniques to identify linear relationships between company information, details of the Chapter 11 case, and the case outcomes. However, modern software makes available a variety of machine-learning tools that can be applied to identify complex patterns in the data and possibly generate better prediction models.

II. The FDJ and the IDB: The Data Used for this Paper

Prior empirical work analyzing Chapter 11 case outcomes has relied primarily on small samples of the available cases because of the difficulty obtaining usable data for the entire set of Chapter 11 filings. For example, the BRD compiled by Professor LoPucki is probably the most popular data set for researching activity in Chapter 11 cases, but it only contains information on about 1,000 cases.[47] Its contents are limited to those cases filed by public corporations with over $100 million in assets (measured in 1980 dollars) since 1979.[48] However, since 2007, over 100,000 Chapter 11 cases have been filed.[49] Further, obtaining data on a national basis has been difficult in the past. Although anyone can go to a bankruptcy court clerk's office and review case information for free using the PACER access terminals, they must have a PACER[50] account to access case information over the Internet.[51] The court charges a fee for each docket or document obtained over PACER,[52] making a large-scale

46. Ellias, *supra* note 3, at 133.
47. *A Window on the World of Big-Case Bankruptcy*, UCLA-LoPucki Bankr. Database, http://lopucki.law.ucla.edu/ [https://perma.cc/U4LE-TVPC] (last visited September 25, 2018).
48. *Id.*
49. *See Statistics From Epiq Systems*, *supra* note 1.
50. *See* PACER, https://www.pacer.gov/ [https://perma.cc/48EL-WN8K]. PACER stands for Public Access to Court Electronic Records and is the system that lets litigants, attorneys, and the public view dockets and filings in federal court cases.
51. *Id.*
52. Currently, documents or dockets cost 10 cents per page, or a maximum of $3.00 a document. *Id.*

exploration expensive. A number of companies assemble documents filed in Chapter 11 cases, but their collections typically exclude some filings, particularly those in smaller Chapter 11 cases.[53] Obtaining large-scale data from these companies for research purposes is also difficult. Finally, extracting structured data from papers filed with the bankruptcy courts, although doable, is a daunting task, requiring significant expertise in natural-language processing systems.[54]

In 2017, the Federal Judicial Center, in conjunction with the Administrative Office of the U.S. Courts (AOUSC), made data available for almost ten years of bankruptcy case filings through its Integrated Database (IDB).[55] For each bankruptcy case, the Bankruptcy IDB provides 126 items of information plus a unique case key. A code book provides details about each item of information in the database.[56] Although the data does not contain debtor names, tax identification numbers, or other personally identifiable information, each record includes the docket number and district, allowing a researcher to look up a particular case on PACER.

53. BUSINESSBANKRUPTCIES.COM, https://businessbankruptcies.com/ [https://perma.cc/B42H-M6FD] (last visited Feb. 22, 2019); *Frequently Asked Questions About Our Services and Database of Chapter 11 Bankruptcy Documents*, CHAPTER11LIBRARY.COM, http://www.chapter11library.com/Faq.aspx [https://perma.cc/FRK2-X3B9] (last visited Feb. 22, 2019); INFORUPTCY, https://www.inforuptcy.com/marketplace [https://perma.cc/XLV4-4Q32] (last visited Feb. 22, 2019); *Overview: Business Bankruptcy Research & Information from BrankruptcyData.com*, BANKRUPTCYDATA, http://www.bankruptcydata.com/p/bankruptcydata-overview [https://perma.cc/BJ85-ZS5F] (last visited Feb. 22, 2019) (providing examples of companies that collect and resell business bankruptcy information).

54. For an example of the techniques involved, see Gunnvant Saini, *How I used Natural Language Processing to extract context from news headlines*, TOWARDS DATA SCIENCE, (Apr. 12, 2018) https://towardsdatascience.com/how-i-used-natural-language-processing-to-extract-context-from-news-headlines-df2cf5181ca6 [https://perma.cc/XL4S-B53D]; *see also* Michael J. Bommarito II, Daniel Martin Katz, & Eric M. Detterman, *LexNLP: Natural language processing and information extraction for legal and regulatory texts*, AIRXIV.ORG (June 10, 2018), https://arxiv.org/abs/1806.03688 [https://perma.cc/Q4WD-Z3FM].

55. *Integrated Database (IDB)*, FED. JUD. CTR., https://www.fjc.gov/research/idb [https://perma.cc/T5RM-GDNP] (last visited Feb. 22, 2019) (retaining bankruptcy data on cases filed, terminated, and pending from FY 2008 to 2017).

56. *See generally* BANKRUPTCY PETITION NEWSTATS SNAPSHOTS DATABASE BPNS DATABASE CODEBOOK, FEDERAL JUDICIAL CENTER (Jan. 2018) [hereinafter IDB CODE BOOK]. The IDB Code Book is available for download at https://www.fjc.gov/sites/default/files/idb/codebooks/Bankruptcy%20Codebook%202008%20Forward%20%28Rev%20January%202018%29.pdf

The substantive information provided for each case is actually very limited. The IDB contains information about case opening and closure activity and final case disposition. For Chapter 11 cases, the IDB provides some summary financial information. However, information from the schedules and statement of financial affairs themselves and information about activity during the case is not available.

For this project, the authors extracted from the IDB information about every Chapter 11 case filed between fiscal year 2008 through fiscal year 2017.[57] When the authors excluded duplicate entries, a total data set of 118,725 cases remained. Some additional cases were removed from the analysis during the data review stage of the project. The authors removed 8,060 cases that were less than two years old that had not yet been disposed. The authors also removed 1,345 cases that were transferred to new districts, filed in error, or dismissed in error, as well as a minor number of additional cases with other data errors. The final data set used included 109,320 Chapter 11 cases. Although the authors removed cases that had been consolidated into a lead case for procedural purposes, cases identified as substantively consolidated cases were left in the dataset.[58]

Some of the information available in the IDB includes:

- The type of debtor;
- The nature of the business;
- Estimated assets, liabilities, and number of creditors, taken from the Chapter 11 petition;[59]

57. *Integrated Database (IDB)*, *supra* note 55. The federal court system runs on a fiscal year ending September 30th. *Id.* To create the IDB data, the AOUSC creates a snapshot of each case filed or open during the prior fiscal year. *Id.*

58. *Id.* About 1,077 substantively consolidated cases remained in the data set. *Id.* This excludes cases jointly administered at the start of the cases and later substantively consolidated. *Id.* One issue with removing substantively consolidated cases is that the Bankruptcy IDB does not directly identify the surviving case when two or more cases are substantively consolidated. *Integrated Database (IDB)*, *supra* note 55.

59. IDB CODE BOOK, *supra* note 56. The Chapter 11 petition requires the debtor to estimate the amount of assets, amount of debt, and number of creditors by selecting from a range of options. *Id.* Each "range" receives a different code in the IDB, and because the selections available change from time to

- Amounts of assets, real property, personal property, unsecured debt, secured debt, and priority debt, each taken from the summary of schedules filed with the debtor's schedules;
- Information about related and consolidated cases;
- Disposition codes, providing information about whether a case was converted or dismissed;
- Chapter 11 percentage dividend, where applicable;
- Chapter 11 future payments.

For the most part, all of the data was used for model development; however, a number of available fields were removed because they were either too sparse to provide usable information for a model[60] or because they were too closely related to matters at the end of the case (and thus inappropriate for use for a model designed to predict end results based on initial case information). In addition to using the filing date for each case, additional features were built based on the month in which the case was filed and the day of the week that the case was filed (for example, a Monday versus a Friday). One feature was the judicial district where a case was filed, and an additional feature included the specific office within a judicial district. Although the data set included the zip code for the debtors' principal places of business, this feature was removed from the dataset to reduce file sizes. However, the zip codes for the county in which the debtors were located were retained as a feature.

Before they could build a model, the authors needed a suitable label or dependent variable against which to train and test the model. The authors wanted to try to predict one of three case outcomes:

time, some revisions to the data were needed to make the options uniform throughout the dataset. *Id.*

60. Sparse data refers to fields where a large percentage of the needed data is missing. Oscar Wärnling & Johan Bissmark, *The Sparse Data Problem Within Classification Algorithms*, (June 5, 2017) (unpublished Bachelor's thesis, Royal Institute of Technology, Stockholm, Sweden), DD142x, http://www.diva-portal.se/smash/get/diva2:1111045/FULLTEXT01.pdf [https://perma.cc/YGP8-PWSB]. The term is also sometimes used to refer to situations where almost all of the data points have the same value. ANDREW GELMAN & JENNIFER HILL, DATA ANALYSIS USING REGRESSION AND MULTILEVEL/HIERARCHICAL MODELS 529–44 (2006).

dismissal, conversion, or viability (defined as a case that either obtained confirmation of a Chapter 11 plan or lasted long enough for plan confirmation to be a viable outcome). A case received the "dismissed" label if one of the dismissal disposition codes was found and the case was still a Chapter 11 case when it closed. A case received the "converted" label if it was no longer a Chapter 11 case when it closed.[61] All other cases were treated as "viable" cases.[62] Of the total data set, 45.14% of the cases were classified as viable, 19.87% were converted to another case under the bankruptcy code,[63] and the rest of the cases were dismissed by the court.[64] Because of the difficulties in testing against three different labels, a second data set was built that combined the converted and dismissed cases into a single "non-viable" outcome, and the potential machine-learning systems were also evaluated using this two-outcome data set.[65]

As a result, the models that the authors built were being designed to test for different outcomes than the Gupta-Barzotto and LoPucki-Doherty models. Those models sought to predict whether a company would be able to continue, post-bankruptcy, as an independent, operating business. The authors' models sought to examine whether a debtor could obtain plan confirmation, either as a stand-alone operating business—as part of a sale or merger—or after a § 363 sale of substantially all of its assets. Prior models have been limited in terms of the amount of data available, as well as the scope of the cases available for examinations; these models were limited to the largest Chapter 11 cases. Although the authors' use of the IDB allows

61. *Integrated Database (IDB)*, *supra* note 55. The authors are aware that many Chapter 11 cases, especially smaller ones, involve a sale of substantially all assets under 11 U.S.C. § 363, after which the case is converted to Chapter 7 to allow a Chapter 7 trustee to complete the remaining tasks in the case, explore preference and fraudulent conveyance claims, and handle final distributions of funds. *Id.* Although these cases could be considered a success, the data in the IDB is not sufficient to identify these situations. *Id.*

62. *Id.* An analysis of the data demonstrated that the two Chapter 11 plan codes only applied to a subset of cases with confirmed Chapter 11 plans and also were apparently not consistently used by all judicial districts. *Id.* The IDB does not contain a specific code indicating whether a Chapter 11 plan was confirmed. *Id.*

63. *Integrated Database (IDB)*, *supra* note 55 (mostly Chapter 7 liquidation cases).

64. *Id.*

65. *Id.*

access to a much larger data set and the ability to examine cases of every size and type, it limits the scope of information available about each case. The IDB does not contain information sufficient to generate the success variable used in the Gupta-Barzotto and LoPucki-Doherty models, and it also does not contain the information needed to engineer the types of features used in those models. This particular project is limited to seeing what kinds of results can be obtained using the Bankruptcy IDB, without significant feature engineering. Better results could almost certainly be obtained by adding the kinds of financial and characteristic information available in the BRD and by engineering additional features.

III. Application of Standalone Algorithms

The authors' initial attempts to build a predictive system using the Bankruptcy IDB data were performed using standard machine-learning algorithms available through the SciKit-Learn library.[66] The three-label data set was used initially, followed by the two-label data set. Using three labels, the authors built models to predict whether particular cases would be dismissed, converted, or remain viable as a Chapter 11 case. Results were judged against a baseline of .45.[67] The use of a label with three categorical outcomes did limit the types of algorithms available. The data was examined using a K-nearest-neighbors (KNN) classifier and a decision-tree classifier. In each case, 20% of the data set was set aside for validating the models.[68]

---

66. *See* SCIKIT LEARN, https://scikit-learn.org/stable/ [https://perma.cc/HA7A-SRRQ] (last visited Jan. 30, 2019). SciKit Learn is a library of classes and functions available within the Python programming ecosystem, which allows the programmer to apply a broad variety of machine-learning algorithms to data. *Id.*

67. Baseline accuracy refers to the prediction accuracy that could be obtained without the use of the machine-learning model, and thus serves as a measure of whether the model is statistically useful. Rama Ramakrishnan, *Create a Common-Sense Baseline First*, TOWARDS DATA SCIENCE (Jan. 12, 2018), https://towardsdatascience.com/first-create-a-common-sense-baseline-e66dbf8a8a47 [https://perma.cc/UJ4K-VGRV]. In this case, if a person simply assumed that all cases were viable Chapter 11 cases, that person would be correct 45% of the time. *Id.*

68. The 20% is referred to as the test data, whereas the remaining 80% of the case data is referred to as the training data. The training data is used to build the machine-learning model, and then the model is

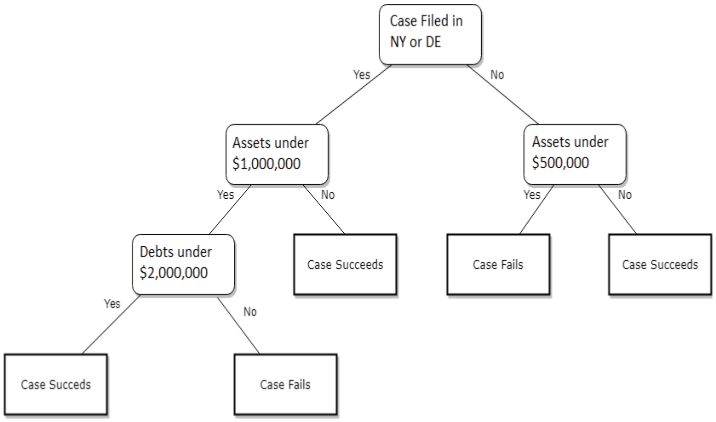
The KNN classifier predicts categories by comparing an unknown data record with similar instances for which the results are known. It operates by measuring similarities between two records based on their features, creating a "distance score" for each set of records within the data set. The algorithm makes a prediction about which class a particular record falls into by looking at the records with the smallest distance scores (its nearest neighbors) and assuming that the record falls into the same category as its neighbors. For example, if the algorithm is set to look at the five nearest neighbors for a particular Chapter 11 case, and three of the five most similar Chapter 11 cases converted to Chapter 7, the algorithm will predict that the Chapter 11 case being tested will also convert to Chapter 7.

An initial model was built using the SciKit-Learn KNeighborsClassifier (a form of KNN classifier) and was set to look at the 100 nearest case filings while weighting those 100 nearest cases based on their distance from the case being tested—in other words, the most similar cases were treated as more relevant. The results on the training set were impressive; the model was able to discern outcomes with 99.9% accuracy. However, the model generated did not do as well on the test set that had been set aside. Accuracy on the test set was only 47%, only slightly above the baseline metric. In short, the algorithm was able to do a good job describing the data used to build a model, but that model failed to generalize to new data.

An alternative model was built that used fewer neighbors but did not use distance weighting. This methodology avoided the extreme overfitting demonstrated by the weighted model while showing similar accuracy on the test data. It had a 57% accuracy on the training data and a 48% accuracy on the test data. However, despite the model's ability to perform better than random guessing, its ability to sort cases into the three categories of converted, dismissed, and viable was not much above the baseline.

applied to the test data to determine how well it performs.

Better results were obtained with a decision-tree algorithm. A decision tree is a branched structure where the branches are controlled by choices made about the features in our data set.[69] The ends of the branches—called leaves—represent the potential outcomes—called class labels.[70] The branches themselves represent conjunctions of features that lead to those class labels.[71] A very simple decision tree might look like this:



At each branch, we examine a different feature variable and make a choice about it—sorting some cases down one branch and the other cases down the other branch. Each data set representing a separate bankruptcy case ends up located in a particular leaf at the end of the decision tree and is assigned a prediction based on the actual results for the majority of the cases that ended up in that leaf. Decision-tree algorithms use statistical measures of accuracy to design the tree to generate the best results. The algorithm determines which data

69. Kevin D. Ashley, Artificial Intelligence and Legal Analytics 110–11 (Cambridge Univ. Press, 2017).
70. *Id.*
71. *Id.*

feature to examine at each branch and the rule that sends a case down one branch as opposed to another.

For this project, the authors built a decision-tree model using the SciKit-Learn DecisionTreeClassifier. Decision trees can easily overfit, and the initial model built was no exception, achieving accuracy of 99.93% on the training data. Efforts were made to generalize the model by reducing the number of splits used by the decision-tree algorithm and preventing splits when the number of samples in a node became too small. The best results were obtained by limiting the number of splits in the tree to twenty levels and requiring a node to have at least 110 cases in it to split further. This produced models with training accuracy of about 70% and accuracy of about 65% on the test data.[72] Although above baseline numbers, this decision-tree model did not obtain high-accuracy results.

Better prediction results were obtained when the system attempted to predict between two outcomes instead of three. To build a two-outcome model, converted and dismissed cases were combined into a single category called non-viable. The models were then trained to predict whether a particular case was viable or non-viable—that is, either dismissed or converted as opposed to the viable category. A decision-tree-classifier algorithm was able to predict between two outcomes with 80% accuracy on the training data and 76% accuracy on the test data.[73] Compared with a baseline of 54.86%, this model showed both better performance over the baseline than the three-outcome model and better overall accuracy.[74]

72. The decision-tree model was tuned using a grid search adjusting both the number of branch splits and number of minimum data points required for a split in order to find the optimum combination of parameters.

73. This model used a maximum of twenty-five layers and required 170 data points in order to split further at a particular point. The model, when trying to predict viability, had on the test data a precision of .75, a recall of .69, and an F-score of .72.

74. A random forest decision-tree algorithm was also applied to the two-outcome data. A random forest decision-tree model reduces overfitting problems by generating a number of decision trees and combining them into a generalized model. Slight improvements were obtained over the standard decision-tree model. A random forest decision-tree model was able to predict between two outcomes with 80.4% accuracy on the training data and 77.3% accuracy on the test data.

IV. The Decision Options Ensemble System and Results

Evaluating potential algorithms to generate appropriate models can be a slow process, and similar or superior results can often be obtained using ensemble systems. When using the techniques discussed above, each type of algorithm has to be evaluated separately. This means building an additional code block to run and test the algorithm and manually adjusting the parameters for each algorithm to identify the combination of parameters that produce the best results. Sometimes changes need to be made to the data set itself to accommodate a particular algorithm. Ensemble systems, which test a number of different algorithms automatically, help avoid this extra work. They also allow the application of multiple algorithms to a particular problem, which can produce superior results.

To test the ensemble methods, the authors deployed a combination of statistical modeling and neural networks to create a model that is able to predict if a bankruptcy case is going to be viable or not. Thus, this model is making a binary prediction.

The authors used an AI platform, Decision Options Technology (DoT),[75] that incorporates over 100 statistical-modeling algorithms and neural-network optimizers. It is able to consume raw data and transform the data to be amenable to modeling in the pre-processing stage. Once the data is cleaned and organized, the platform can conduct feature engineering—that is, select the attributes that are most valid for the problem being solved. In large datasets, it also does sampling with an objective of no information loss. These steps reduce the amount of data that needs to be fed into the modeling process and allows the system to quickly evaluate a large number of possible algorithms. We then optimize these engines by fine-tuning parameters that control the algorithms. For example, the neural net could have many different layers, and the number of neurons in each layer can vary. Additionally, convergence of the modeling process depends on the initial conditions specified as well as the optimizer

---

75. *See generally* DECISION OPTIONS, http://www.decisionoptions.com/ [https://perma.cc/RGC6-MTCH] (last visited Feb. 22, 2019) (DoT is a product of Decision Options).

used. Thus, for the selected mathematical techniques, we create hundreds of models that differ in their configurations, each producing a different level of confidence and robustness. The technology works from the cloud and uses heavy, parallel processing to create usable models within the allocated time.

During the modeling phase, the authors ran statistical algorithms and neural nets and selected those showing the highest robustness. In making the selection of the best approaches, the authors used what is called a "[ten-fold] cross-validation."[76] One issue with ensemble modeling is that it is easy to overfit the data—that is, the system can make models with very high confidence using the data presented but fail when used on new data. These models will not be useful in practice. To avoid this, we take the entire dataset and divide it into ten different pieces. We take extreme care not to have duplicates in these ten, mutually exclusive bundles of data. Then we create a model taking nine of the data bundles described to make the model and then test the model on the remaining tenth bundle. The confidence of this model is indicative of robustness as we are testing on unseen data. We repeat this process ten times, each time making a different bundle to be the test bundle and the remaining nine used in training the model. We then average the confidence levels across the ten models to get an estimate of expected confidence of the mathematical approach if used on unseen data.

Once the authors selected the best mathematical technique by comparing hundreds of techniques using the cross-validation process described above, they generated models using these techniques, each able to make predictions at different confidence levels. In the problem described in this paper, the authors ended up with eighteen different techniques; some statistical and others neural net based. The authors created an ensemble model by wrapping these models using a weighted-voting mechanism.[77] That is to say, the ultimate predictions

76. ASHLEY, *supra* note 69, at 113.

77. *See generally* D. Optiz & R. Maclin, *Popular Ensemble Methods: An Empirical Study*, 11 J. ARTIFICIAL INTELLIGENCE RES. 169 (1999).

made by the ensemble model is a sort of weighted consensus of the eighteen models that passed the robustness test. This allows us to further enhance the usefulness and validity of the modeling process.

The ensemble model described shows about 75% confidence in correctly predicting for both classes. We see approximately the same confidence level in random split hold-out data. Because the confidence level is produced by cross-validation, we expect this to be the case if the model is applied on newly arriving data. This means that if it were to predict if a case is viable at inception using the characteristics of the case (when the outcome was not known), we would be correct three-fourths of the time. Because the model will make a probabilistic prediction for a new case—that is, it will give a probability that the case will be viable—the results can be further interpreted in practice.

Analysis

The results from this project provide a number of insights to guide future activity in building prediction systems for Chapter 11 cases. Both the manual decision-tree model and the ensemble model generated similar accuracy results, showing a demonstrated ability to correctly identify whether a particular Chapter 11 case is viable about 75% of the time. This result is significantly better than the baseline of 55%, which someone may achieve by simply assuming that all cases fail or by guessing randomly. On the other hand, model accuracy is substantially below that claimed for the Gupta-Barzotto model discussed earlier. This is possibly because accuracy numbers for that model were only reported for the training information or possibly because of the richer information available in the BRD compared with the Bankruptcy IDB.[78] On the other hand, the models described here can be applied to all bankruptcy cases, not just the large capital cases tracked by the BRD.[79] The larger dataset used also allows for

78. Gupta & Barzotto, *supra* note 15, at 31.
79. *A Window on the World of Big-Case Bankruptcy*, *supra* note 47.

more statistically relevant results in addition to allowing researchers to employ methods, such as neural networks, that cannot generate reliable models with small datasets.

Another question worth considering is how the models described in this paper may compare with or supplement current practices. In many Chapter 11 cases, attorneys and courts make decisions about potential viability by reviewing the relevant information available in the case and qualitatively evaluating the case's prospects. Sometimes these decisions are easy ones. For example, assume that the debtor operates a retail store, the lease was terminated pre-petition, and the landlord is asking the court for permission to evict. Here, predicting a conversion or dismissal is easy. In many other cases, however, deciding whether a case will succeed is very difficult using qualitative methods. Quantitative financial analysis is often employed to ascertain potential outcomes, but the courts themselves are limited to the analysis provided to them by the parties, and—especially in smaller Chapter 11 cases—the parties lack access to financial professionals capable of doing an adequate job using quantitative methods. In any case, financial professionals of case parties are often tasked with providing a quantitative basis to support a particular outcome; these analyses are valuable but not statistically relevant. Statistical models can certainly supplement other techniques, as well as provide a mechanism for decision making using smaller information sets.

Ideally, the techniques discussed in this paper could be applied to richer information than what is currently available through the IDB. This could include financial information from publicly available sources or historical company financial statements, information extracted from the petition, schedules, and first-day pleadings (possibly using natural language processing techniques) and docket information. However, building useful models requires collecting relevant information for a large set of prior cases, not just the case being examined. The BRD does this currently for a relatively small number of cases.[80] However, expanding this information collection to the complete corpus of Chapter 11 cases presents practical issues, mostly how to avoid the economic cost of obtaining court documents through PACER.[81]

Even so, models like the ones described in this paper will provide tactical and strategic advantages for the stakeholders that employ them, as well as improve decision-making. The techniques described can also be used to predict other aspects of the bankruptcy process such as short-term outcomes, professional fees, and distribution results. They can also be applied to other Chapters of the Bankruptcy Code.[82]

Decision models like these are not static. Typically, these models are built as learning models and are able to self-learn and retune as new data with known outcomes become available. If systematically deployed in a decision process, these models can enhance human judgment and intuition. Further, model behavior over time will indicate changes that may be driven by regulations, behaviors, and other structural changes, or even changes in decision-making behavior driven by the model's use. Building usable models requires access to large datasets as well as a combination of domain knowledge and data science expertise not yet widely available in the legal industry. However, AI and machine-learning-based models can provide dynamic and statistically accurate results to support decision-making in ways not currently available.

80. *Id.*

81. The Electronic Court Records Reform Act of 2018, introduced September 6th, 2018, would eliminate PACER access costs, greatly reducing the cost of assembling court information for modeling use. Jason Tashea, *Proposed Legislation Would Eliminate PACER Fees*, ABA J. (Sept. 18, 2018), http://www.abajournal.com/news/article/new_bill_wants_to_end_pacer_fees [https://perma.cc/2P7W-RRJM]. However, it does not appear to be making progress in the House of Representatives. *Id.*

82. *See generally* Warren Agin, *Using Machine Learning to Predict Success or Failure in Chapter 13 Bankruptcy Cases*, 2018 ANN. SURV. OF BANKR. L. 13 (2018).

Faculty

Warren E. Agin is managing director of Digital Strategy & Solutions at Elevate Services in Boston and a member of its Global Leadership Team. He provides subject-matter expertise, builds machine learning systems, and consults in the areas of data analytics, application of artificial intelligence, and digital transformation. He has also published multiple papers on the application of machine learning techniques to legal case outcomes in the bankruptcy field. Previously, Mr. Agin was part of Elevate's data science group, which provides uniquely qualified services in legal analytics, legal data science and training, risk management, and legal data strategy consulting. He also founded and currently chairs the American Bar Association's Legal Analytics Committee, which helps business lawyers understand how to use artificial intelligence and other analytic techniques. A practicing lawyer for almost 30 years,Mr. Agin has taught legal analytics as an adjunct professor at Boston College Law School and serves on the Governing Council for the ABA's Center for Innovation. Prior to working with Elevate, he was a partner at Swiggart & Agin, LLC, a corporate and bankruptcy boutique he co-founded in 2000, and he served as a chapter 7 bankruptcy trustee. Mr. Agin has authored *Bankruptcy and Secured Lending in Cyberspace, 3rd Ed.* (West 2007) (the first book to discuss the use of technology in bankruptcy practice) and the *Bankruptcy and Intellectual Property Deskbook* (ABA 2016), and he has frequently lectured on using machine learning and data in legal practice for groups such as the American Bar Association, the National Conference of Lawyers and CPAs, Massachusetts Continuing Legal Education, Thomson Publishing and the Institute of Legal Information Theory and Techniques. He received his undergraduate degree in business administration with a minor in economics from Washington University in St. Louis and his J.D. from Boston College Law School.

George Sanderson is the principal and founder of the Sanderson Law Firm, PLLC in Raleigh, N.C., and advises individuals and companies on a wide array of business bankruptcy and business-litigation matters. Before starting his own firm, he was a partner in one of the top boutique law firms in North Carolina and had been associated with two other nationally recognized law firms. Mr. Sanderson has appeared in courts and arbitration tribunals throughout the country. His bankruptcy experience includes representing debtors, creditors, officers, directors and trustees in numerous corporate chapter 11 and 7 bankruptcies. He has counseled clients on restructurings, loan workouts, asset purchases and sales, lease assumptions and rejections, preferences, fraudulent transfers, vendor claims, officer and director liability, and lender liability. Mr. Sanderson is currently the chair of the E-Discovery Subcommittee of the ABA Business Law Section's Committee on Bankruptcy Court Structure and Insolvency Process. He was named to *Business North Carolina*'s Legal Elite Hall of Fame for his bankruptcy work in 2020, and he has also served as the chair of the North Carolina Bar Association's Antitrust and Complex Business Disputes Section. Mr. Sanderson received his B.A. *magna cum laude* from Georgetown University and his J.D. from New York University School of Law, after which he clerked for Hon. Stephen C. St. John of the U.S. Bankruptcy Court for the Eastern District of Virginia.

Kristi R. Smith is the senior manager of Innovation Solutions at Bryan Cave Leighton Paisner LLP in St. Louis and serves as a member of the firm's Global Innovation Team. As a licensed attorney with an aptitude for incorporating innovation into legal processes, she specializes in utilizing expert systems, automation, intelligent workflows, collaboration tools, and other technologies to dramatically improve the efficiency and effectiveness of legal service delivery for the firm's lawyers and clients.

She also designs and architects solutions for the firm's new division, BCLP Cubed, which provides clients with efficient, process-driven volume legal delivery of high-quality complex legal advice, best-in-class technology and experienced legal operations support. Ms. Smith regularly speaks and authors on topics such as innovation in diversity and inclusion, artificial intelligence in the law, improving access to justice, and the future of lawyering. She is also active in the innovation community, both locally and worldwide, with a passion for creating new ways to connect those who have historically been precluded from access to justice with the right tools, resources and expertise committed to improving social equality. Ms. Smith is a member of the Bar Association of Metropolitan St. Louis and the International Legal Technology Association. She received her B.A. in 2006 from Ouachita Baptist University and her J.D. in 2010 from Saint Louis University.

Nellwyn W. Voorhies is executive director of Donlin, Recano & Company, Inc. in San Diego and has more than 25 years of experience in the restructuring legal community. She previously practiced at several large firms, including Levene, Neale, Bender, Rankin & Brill, LLP, Baker & McKenzie and Sheppard, Mullin, Richter & Hampton LLP. Ms. Voorhies brings her legal expertise to the claims management industry, having been a director at Kurtzman Carson Consultants LLC and vice president of Sales and Marketing at Rust Omni. She worked on numerous chapter 11 cases as an attorney, including Daewoo Motor America, Stateline Hotel Inc., C&R Clothiers and Kenny Rogers Roasters. Ms. Voorhies serves on ABI's Southwest Bankruptcy Conference advisory board, is active within the American Bar Association's Business Bankruptcy Committee of its Business Law Section, and serves on the board of the International Women's Insolvency & Restructuring Confederation (IWIRC). She received her B.A. *cum laude* from Georgetown University and her J.D. from the University of California, Berkeley.