Proactive Collections Management: Using Artificial Intelligence to Predict Invoice Payment Dates

By: Sonali Nanda

As originally published in the Credit Research Foundation 1Q 2018 Credit & Financial Management Review

Abstract

Machine Learning, being the hottest trend today, is incredibly powerful for predictions or calculated suggestions based on large amounts of data. As businesses from many industries have begun to rely more heavily on machines to do the heavy lifting of A/R processes, executives are able to deliver more value to their customers by reflecting on their roles and identifying the opportunities machine learning could offer them.

Executive Summary

The cut-throat nature of businesses today requires firms to adjust their strategies and apply financial policies to survive and enable growth. Accounts receivable (A/R) represents the dollar value of business that a company has transacted for which it has not yet received payment. It appears on the "assets" side of a balance sheet; however, more often than not, A/R threatens complex financial difficulties and choked cash flow. This article investigates the problem of reducing outstanding receivables through improved collections strategies. It illustrates how Machine Learning and Artificial Intelligence, using historical data from accounts receivable, could be leveraged by A/R financials to make predictions that are four times more accurate than traditional estimation metrics such as Average Days Delinquent (ADD), which in turn helps fine-tune the collections process.

Traditionally, business executives across the financial spectrum believed that Machine Learning and Artificial Intelligence were just millennial buzz words. This assumption could not be more flawed. Today, Machine Learning and Artificial Intelligence are the top trending elite technologies available to ensure that collections activities are more effective, starting at root operational levels so that collections efforts are optimized for maximum productivity with minimum cost.

A typical collections process is largely reactive and relies heavily on due dates as the pivot for all dunning activity. It starts only when an invoice is due or shifts to a larger aging bucket. The majority of collections operations, including account prioritization, correspondence strategies, and customer collaboration, are based on static parameters such as aging bucket and invoice value. This results in a cluttered collections worklist, inefficient identification of delinquent accounts, and wasted collections efforts. Due to the absence of a scalable collections process, which takes dynamic parameters into account, the collections team ends up chasing only past due A/R while the overall team productivity is lost in labor-intensive, time-consuming, low-value tasks such as ERP data extraction, manual worklist creation and correspondence with non-critical customers. The key fallouts include a slower cash conversion cycle, increasing DSO, inefficient processes and higher operational costs.

The dynamic shift from a reactive to a proactive collections process is one of the beginning perks of an AI-powered collections management process. With machine learning under the hood, the collections team could leverage high-impact predictions to enhance collections output and key KPIs such as DSO (Days Sales Outstanding) and CEI (Collection Effectiveness Index). Predicting payment date and delay could be the foundation of a modern-day collections process, which takes the dynamic changes in customer behavior into account when formulating dunning rules and strategies. Further, customer collaboration could be tailored and personalized by analyzing customer preferences in terms of time, day of the week, and mode preferred for communication and with insights on identifying which dunning letters work best for each customer.

The State of Collections Today

Disassembling Collections

Collections is one the most important processes in A/R. For a business to have a leading growth graph, the A/R team needs to put effort into Collections Management. Over the years, collections management has evolved from a call-centric “dial for dollars” process to a process with a versatile set of operations, including worklist prioritization, customer collaboration, logging call notes and communication details, saving and tracking payment commitments, setting up reminders and follow-up, and monitoring collections analytics. A typical collections process is illustrated below:

- **Worklist Prioritization**
  The first step in the collections process is to create a prioritized worklist. This worklist determines the order in which the collector will work or initiate the dunning process on the open or past due invoices. Conventionally, A/R teams have considered passive factors including aging and invoice value to prioritize accounts. However, these lagging factors of delinquency have not yielded satisfactory output over the decades.
- **Customer Collaboration**
  The collectors contact customers based on the prioritized worklist. The cornerstone of the collections process, correspondence, relates to any activity that is performed when reaching out to a customer for payment. This includes calls, sending reminders, past-due notices, demand notices and sharing account statements, invoices, and any other document that is required by a customer.
- **Logging Correspondence**
  Collectors need to keep track of all the information shared during correspondence. This could include logging call notes, storing payment commitments, noting down specific queries made by the customer, and adding any other future follow-up tasks to be performed in the worklist. This activity not only helps the collectors stay on top of their accounts by having all previous communication details in one place, but also ensures seamless transition of accounts within the collections team, especially in case of escalation.
- **Tracking Payment Commitments and Setting Reminders and Follow-Up**
  Customers often inform collectors about payments that they are going to make in the future. However, the obligation of keeping track on whether these commitments have been honored or not lies with the collector. Collectors need to keep a close watch on all commitments made by customers and follow up if the payment has not been made.

- **Reporting and Analysis**
  To monitor the performance of the collections management process, it is essential to regularly analyze the deliverables of the team, the input costs and the output of collections efforts. This helps to identify the strategies which work best and drive positive outcomes, and the poor performing tactics which should be weeded out. Moreover, monitoring performance provides insights to enhance long-term strategies and make the collections process scalable.

The Four Pillars of Collections

In essence, an ideal collections management process rests on four pillars:

1. **Data:** Data is the backbone for all collections operations, including worklist prioritization and customer correspondence. This includes all of the information extracted from the ERP, the data obtained from other teams such as cash application, credit management, deductions management, billing, and A/P, and the information gathered from the customer such as payment commitments and invoice discrepancies. Real-time and accurate data is the most basic element in all collections activities.
2. **Customer Collaboration:** Undoubtedly, customer collaboration is the heart of the collections process. But isolating the at-risk, critical customer from the fast-paying, low-risk customers is no mean feat and has a significant impact on the outcome of the customer correspondence and account coverage for a collector. Automated correspondence for low-risk customers is the simple yet impactful answer to enable collectors to focus on crucial accounts.
3. **In-Line Correspondence Logs:** With an exponential increase in collections workload, it is not feasible for collectors to keep track of all previous customer collaboration through notepads or spreadsheets. It is essential to log correspondence within the collections management tool to reduce the pre-correspondence workload for the collectors.
4. **Account Prioritization:** With tens of thousands of customers and a multitude of open invoices, spending time and efforts on critical collections accounts is of paramount importance for every collector to ensure account coverage and positive growth for collections closed. In this scenario, relying on static parameters such as due-date for prioritization, as discussed before, is simply not an option for collectors. The collections teams across the industries need to consider dynamic, leading indicators as the bedrock for worklist prioritization for a scalable collections process.

Where the Collections Shoe Pinches

As much as the collections process has evolved, some of the pitfalls in the process have failed to disappear. Collectors still struggle across the roadblocks such as too many delinquent accounts and a shortage of time to cover all accounts in the worklist. The following are fundamental cracks in the pillars of an ideal collections process:

1. **Inaccurate and Stale Data**
   Data serves as the fuel to the collections engine, and inconsistency in data severely impacts the collections process. For instance, collectors often hear from the customers during a call that they have already made payments against an invoice in question. This

scenario occurs when the cash is applied in batches and the real-time updated information is unavailable to the collectors, or when the cash has been applied incorrectly. This not only wastes the collector's time, which could have been used for an at-risk account, but also renders the invested collections efforts redundant and presents poorly to the beleaguered customer.

2. **Half-Baked Correspondence**
   Executives across the financial spectrum believe that collections correspondence is the core of collections. As golden as this belief is, the fact still remains that the groundwork beneath customer collaboration is more pivotal. Answering "who to contact?" is much more crucial than "how to contact?" Collectors often do not focus on the groundwork and end up focusing on non-critical customers who would have paid even in the absence of frequent follow-up and reminders.
3. **Static Prioritization**
   At the outset, the collectors relied on their intuition, skill, and experience to skim through the worklist in to order to prioritize accounts and reach out to the customers. As collections management has developed into a reactive process, the collectors use static parameters such as due-date, invoice value, and current customer segment to tweak their collections strategies. With the numerous factors involved in a customer's A/P functions, these factors are grossly insufficient to deal with dynamic changes.
4. **Typical Solutions and Their Achille's Heel**
   While using technology to automate the collections process is the current trend, a misguided decision on the "scalable" solution could become the Achille's heel for a company's receivables. An ideal, scalable solution provides an efficient, compliant and cost-effective process by optimizing the four basic factors of an automated collections process.

Typically, the automation for each of the four keys pillars of the collections process is:

1. **Real-time Data:** Most of the trending collections automation solutions provide real-time data integration to the A/R teams, ensuring availability of data that is accurate and available virtually immediately at their fingertips.
2. **Automated Correspondence:** With the progressive drive towards e-adoption and digital enablement of the customers, automated correspondence is one of the most common features offered by top software available in the market today. Automated dunning via email, fax, print and mail with easy-to-create correspondence templates is an effortless recipe to improve account coverage, achievable through technology solutions currently available.
3. **Log Correspondence in a Single System:** Most automation facilitates call notes, correspondence logs, and other means to document communication with the customer within the solution itself. This eases the workload of collectors and ensures a single-point access to all communication history with all customers.
4. **Prioritization - Still Stagnant:** Regardless of the challenges and roadblocks offered by static prioritization, most automations are still dependent on lagging factors, such as due-date and invoice value, to slice and dice the invoice data extracted from the ERP. While this technique may seem reasonable and insightful enough from a top-level view, the industrial statistics and leaders disagree. Static prioritization is the core of a reactive

collections process which ensures that dynamic changes in customer behavior are obscured in worklist prioritization. Teams that have understood the importance of proactive collections and dynamic parameters have tried leveraging Average Days Delinquent (ADD) as a predictor and leading indicator but failed to produce noteworthy results.

The Problem with Averages

What is Average Days Delinquent (ADD)?

Average days delinquent (ADD) is the average number of days that invoices are past due - the amount of time between the invoice due date and the date it is paid. This calculation helps a company evaluate, along with other factors, the overall performance of collections department and their ability to convert accounts receivable to cash.

HOW TO CALCULATE AVERAGE DAYS DELINQUENT

1. Calculate average Days Sales Outstanding (DSO)
   DSO = (Average AR / Total Credit Sales) x Number of Days

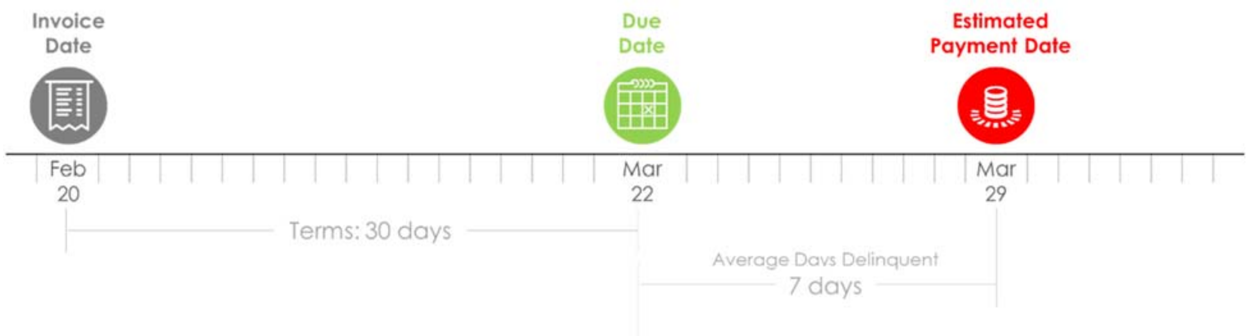
2. Calculate Best Possible DSO
   Best Possible DSO = (Current AR / Total Credit Sales) x Number of Days

3. Calculate Average Days Delinquent
   ADD = Days Sales Outstanding – Best Possible Days Sales Outstanding

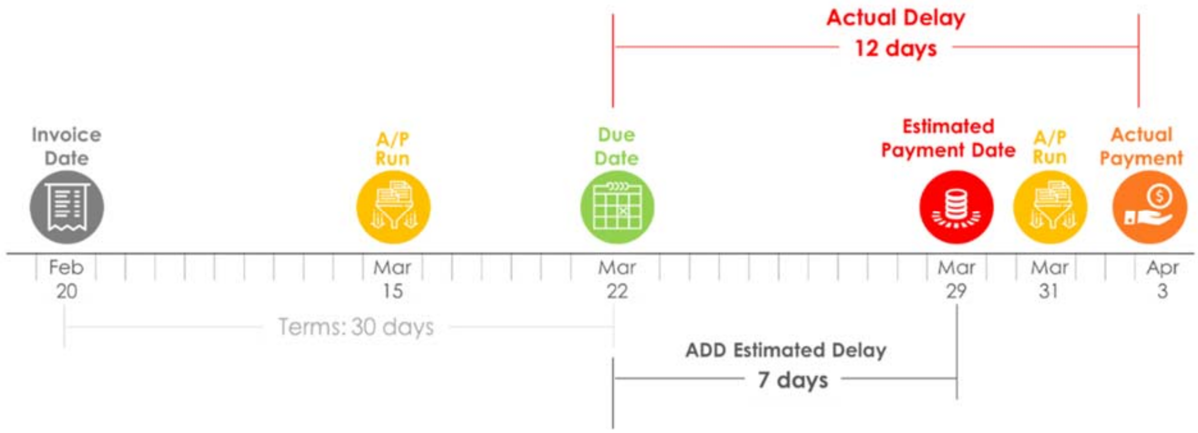
Prediction with ADD: Fundamentally Flawed

In an attempt to unlock the strategic benefits of a proactive collections process, collections teams regard ADD as the current best metric to estimate payment date for a customer and consequently implement dynamic strategies and rules for proactive correspondence.

The subsequent figure illustrates payment date prediction using ADD as a metric for a small-to-medium business (SMB) customer where the A/P team runs its cycle in the middle or end of the month and the payment terms are 30 days. This is just one potential pattern with one customer where the A/P cycle plays a vital factor in payment.

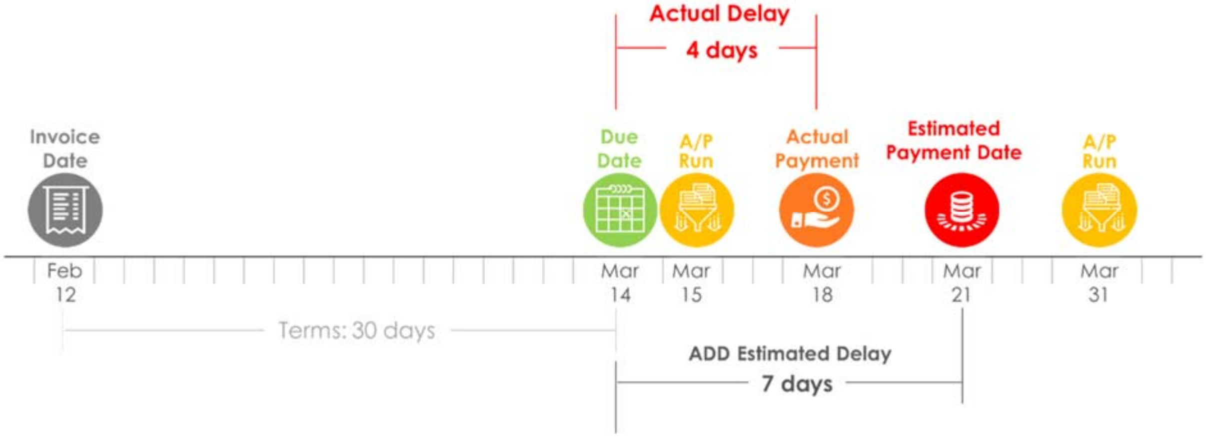


In the above scenario, the invoicing date is February 20th, and the due date is March 22nd according to the 30-days payment term. The ADD value of 7 indicates that on an average, the customer delays the payment by 7 days after the due date. Based on this estimation, the predicted payment date is March 29th. However, the A/P cycle schedule has not been considered in this assessment.



The dynamics of the real-world overshadow this prediction. The A/P cycle of the customer runs on March 15th, i.e., mid-month when the payment terms are still valid and the payment is not due, and therefore, the A/P team skips initiating the A/P process for this invoice. The next A/P cycle runs on March 31st when the invoice is already past-due. This is when the A/P team begins the invoice approval and payment process, and the actual payment comes in on April 3rd. As a result, the actual delay is 12 days as compared to the 7 days prediction delay.

In another scenario, the invoicing date is February 12th, and the due date is March 14th, while the predicted payment date, based on ADD of 7 days, is March 21st.



In this case, the A/P cycle runs on March 15th, one day after the due date, and approves the payment for the invoice. As a result, the actual payment is made on March 18th with a delay of 4 days, as compared to the predicted payment date of March 21st with a delay of 7 days.

The above scenarios clearly demonstrate that ADD as a metric is not sufficient to predict payment date and needs other customer-specific factors, such as A/P cycle schedule, to predict the payment accurately. Moreover, the above represents an instance of a single customer. With the gigantic number of customers with which the collectors deal, a nearly infinite number of factors come into play. It is not practically feasible for collections teams to identify each influencing factor and corresponding pattern for each customer to predict payment date and tweak collections strategies.

Why Machine Learning?

Evolved from the study of pattern recognition and computational learning theory in Artificial Intelligence, Machine Learning explores the study and construction of algorithms that learn from and make predictions on large volumes of data. It is the science of getting automations to act without being explicitly programmed to do so.

Machine Learning could be leveraged to enhance collections process as it enables payment date predictions that are up to four times as accurate, using historical A/R data. As discussed in the previous section, machine learning identifies relevant variables and analyzes valuable patterns in the collections cycle to make an educated guess on the payment date for each customer - science that is practically impossible for humans. Machine Learning has the ability to process, analyze, and identify patterns amidst the enormous volume of historical data available for each customer. It is able to predict the payment date at an invoice level for all customers and help the collections teams become proactive through improved dunning strategies.

Embedding ML in Collections

The ML Toolbox

Machine Learning being the hottest trend today, is incredibly powerful for predictions or calculated suggestions based on large amounts of data. As businesses from many industries have begun to rely more heavily on machines to do the heavy lifting of A/R processes, executives are able to deliver more value to their customers by reflecting on their roles and identifying the opportunities Machine Learning could offer them.

This section highlights the key algorithms explored for payment date prediction in the collections process. The following models were considered and evaluated for the same.

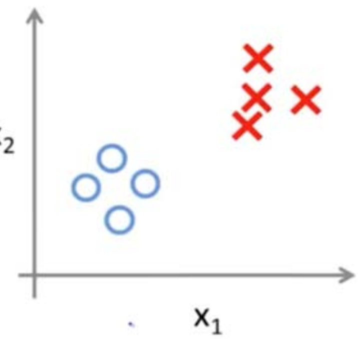
- **Linear Regression:** Linear regression is a linear model, i.e., a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that (y) could be calculated from a linear combination of the input variables (x).
- **Logistic Regression**: Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables which determine an outcome. The outcome is measured with a dichotomous variable (one with only two possible values).
- **Decision Tree**: Decision Trees are algorithms where the data is continuously split according to a certain parameter. The tree is explained by two entities called decision nodes and leaves. The leaves are the decisions or the final outcomes, and the decision nodes are where the data is split.
- **Support Vector Machine (SVM):** A Support Vector Machine (SVM) performs classification by finding the hyperplane that maximizes the margin between two classes. The vectors (cases) that define the hyperplane are the support vectors.
- **Naive Bayes:** In Machine Learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.
    - In probability theory and statistics, Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event.
- **k-nearest neighbors (KNN):** K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).
- **Random Forest:** Random forests or random decision forests operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
- **Gradient Boost & Adaboost:** These are boosting algorithms. Ada(Adaptive)boost(Boosting) is an iterative process that fits a sequence of weak learners on different weighted training data. It starts by predicting an original data set and gives equal weight to each observation. If the prediction is incorrect using the first learner, then it gives higher weight to observations which have been predicted incorrectly. Gradient boosting minimizes the loss of the whole system using the Gradient Descent method.

- **K-Means:** K-Means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K.
- **Dimensionality Reduction:** In Machine Learning, dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration by obtaining a set of principal variables. It could be divided into feature selection and feature extraction.

The following describes some models in detail and explores how they integrate with the collections process.

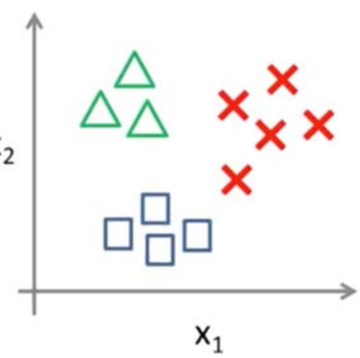
1. **Binary Classification Model**
   Binary Classification is the task of classifying the elements of a given set into two groups (predicting which group each one belongs to) on the basis of a classification rule.

   In terms of collections, this model predicts delay by answering Yes or No to the question: "will payment for a given invoice will be delayed?"



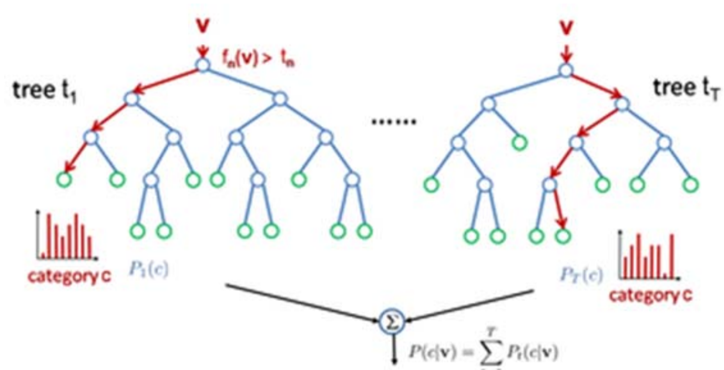
2. **Multiclass Classification Model**
   In Machine Learning, multiclass or multinomial classification is the task of classifying elements of a given set into one of three or more classes on the basis of a classification rule.

   For payment date prediction, this model classifies invoices into three or more buckets based on the number of days an invoice is delayed.



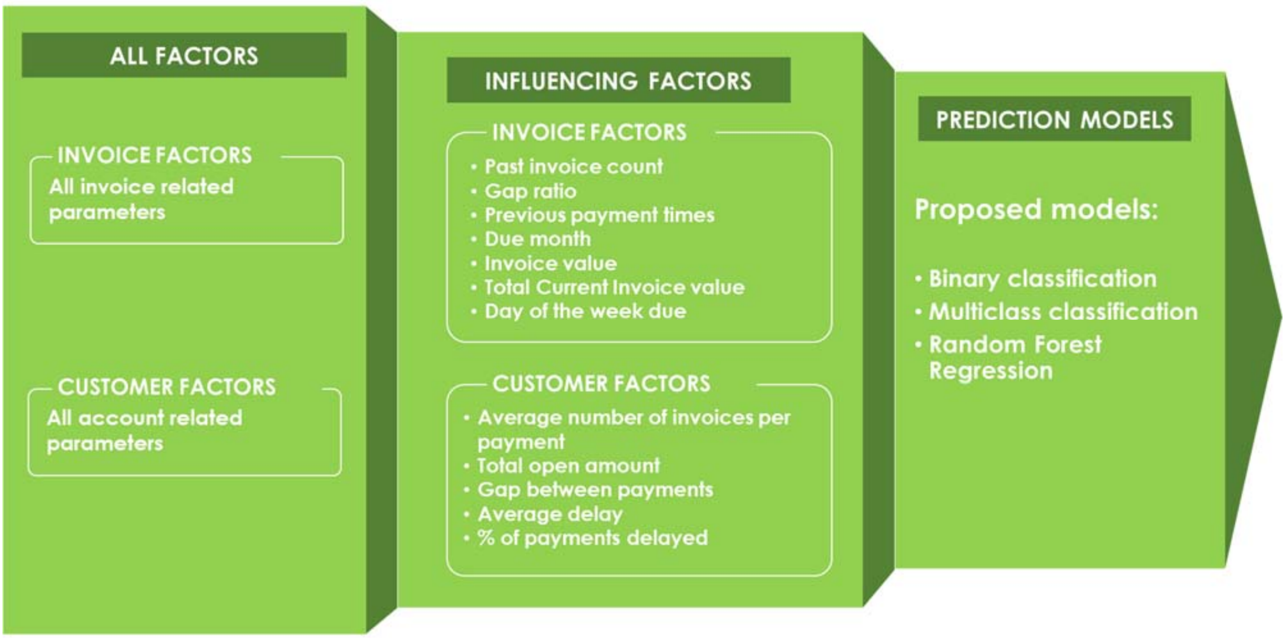
3. **Random Forest Regression Model**
   In machine learning, a regression model is used to predict a continuous-valued output.

   This model predicts an actual payment date based on the features such as delay ratio (Number of Delayed Invoices/Total Number of Invoices) and their importance.



Features in Play

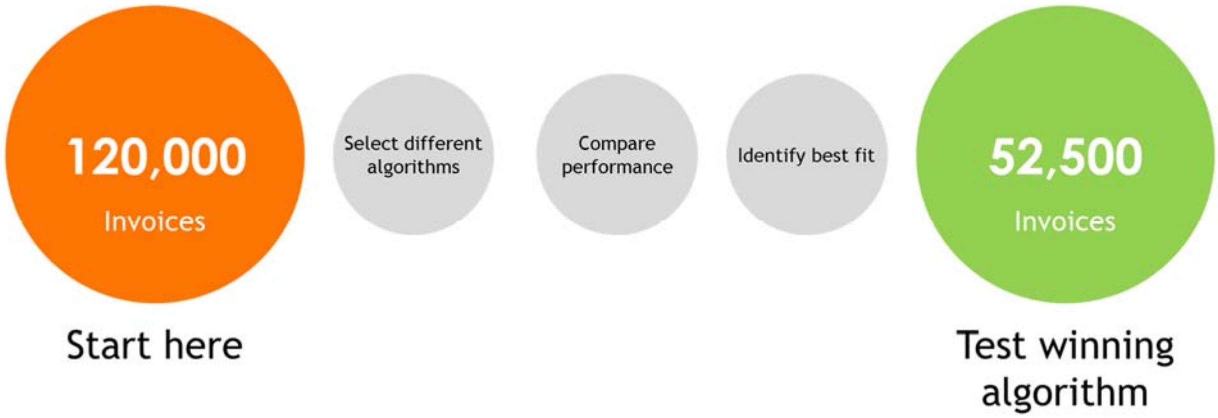
In Machine Learning, "features" refers to the factors that influence the outcome or prediction results. With respect to collections, these features include the invoice level and customer level details that play important roles in identifying the payment trends for each individual customer.



Training and Testing for Machine Learning

In Machine Learning, a model is generated to predict the results based on test data. The training data is used to analyze information and identify patterns from the training data, i.e., to fit the model while testing data is used to test the model in terms of accuracy and precision. In layman's terms, training a model is when the inputs and outputs are known (the training dataset) and are used to identify the computation behind the transition from input to output. Testing a model is when the input (testing dataset) is known and computed by the model to predict the output. This predicted output is compared to the real output (testing dataset) to evaluate the accuracy in prediction.

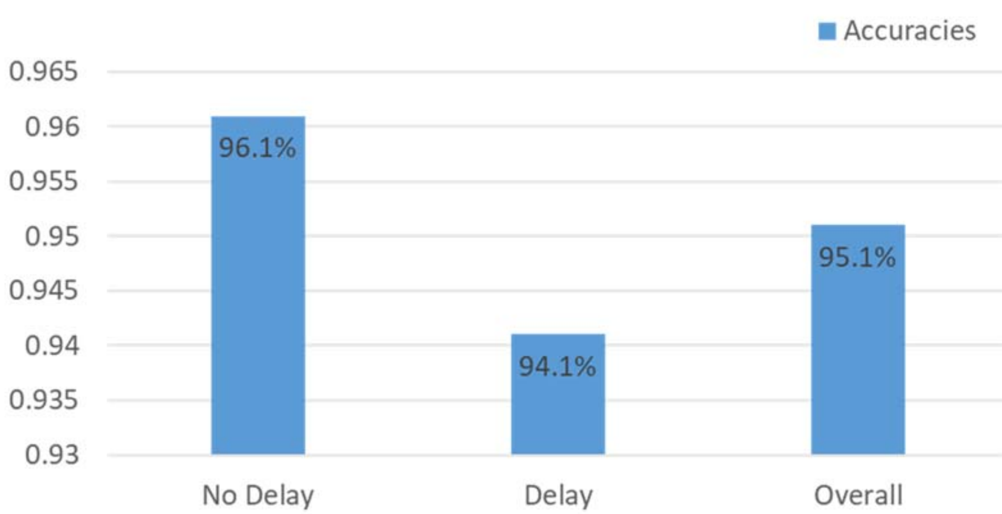
For implementing Machine Learning in collections, the following methodology was used to generate models, their training and evaluation through testing.



As discussed earlier, classic models were considered and trained based on invoice and customer level features. The subsequent section illustrates results and important features for some models.

1. **Binary Classification Model**

   Result:

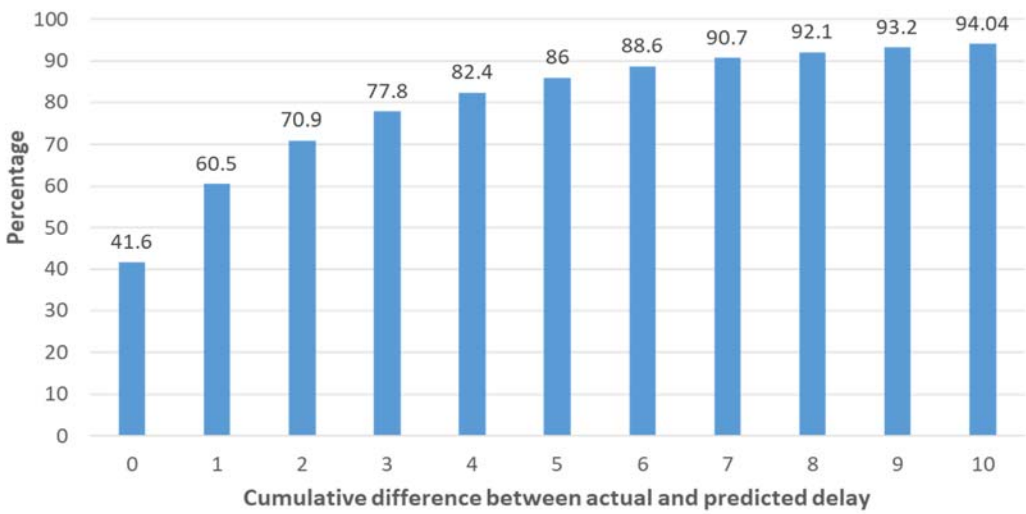


2. Multiclass Classification Model

Result (Accuracies):

3. Random Forest Regression Model

Result (Percentage of Invoices Predicted Correct [cumulative]):

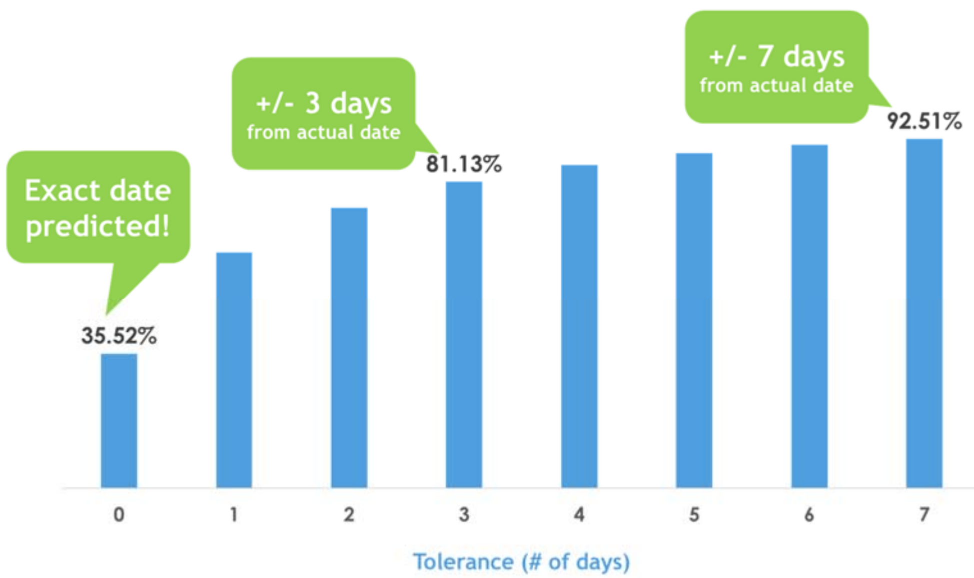


The Chosen One – Random Forest Regression

As mentioned in the previous chapters, the collections process encompasses numerous factors (features) which could be leveraged to identify the payment trends of each customer and make predictions based on these features. In Machine Learning, regression analysis is used to predict the exact (continuous) value rather than a class (range of values) prediction as with classification algorithms. With respect to collections, regression analysis predicts an exact payment date while the classification algorithms predict the payment delay in terms of buckets such as [0,5], i.e., delay between one to five days.

Since the collections process has a high number of predictive variables and a huge sample size, the random forest model was an ideal fit. This is because it captures the variance of several input variables at the same time and enables a high number of observations to participate in the prediction.

The Random Forest Regression Model was identified as the ideal model which could predict the exact payment date for each invoice with higher accuracy as illustrated in the graph below:



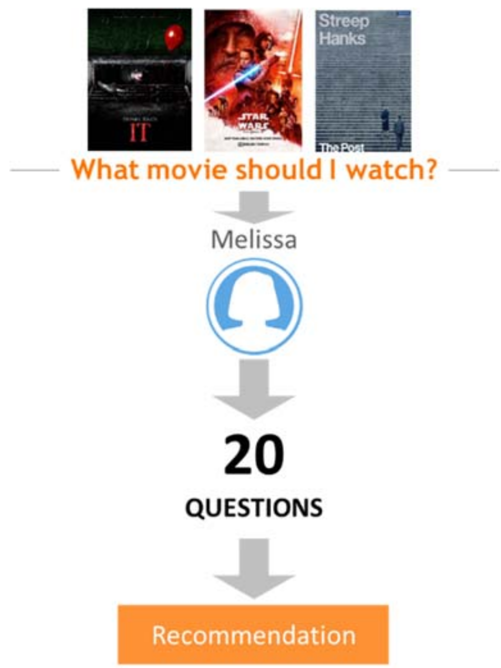
The next section explains the working of a random forest regression model.

An A/R Guide to Random Forest Regression Model

The Random Forest Model

A Random Forest consists of an ensemble of an arbitrary number of simple tree predictors which are used to determine the final outcome, each capable of producing a response when presented with a set of predictor values. In a regression problem, their responses are averaged to obtain an estimate of the dependent variable (output) given the predictors.

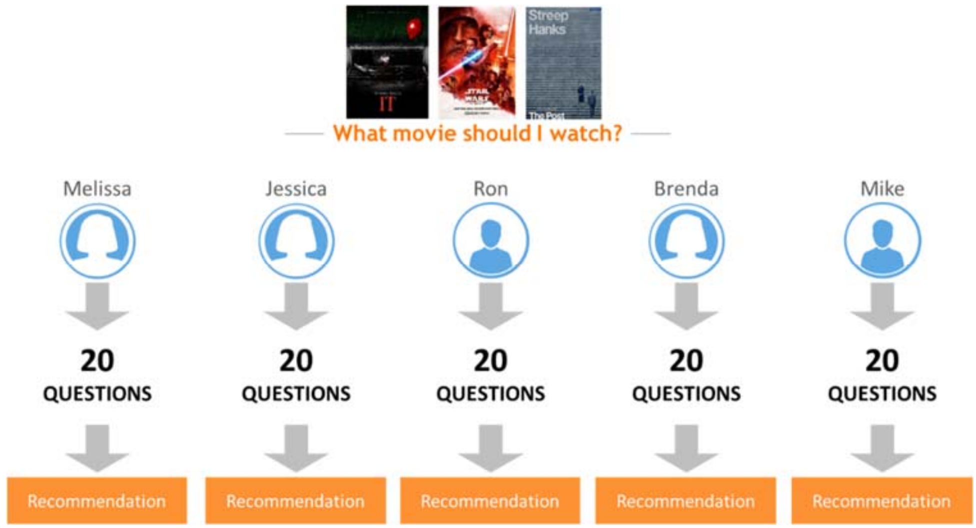
This section describes the random forest model in layman's terms by drawing an analogy to a simple real-life scenario.



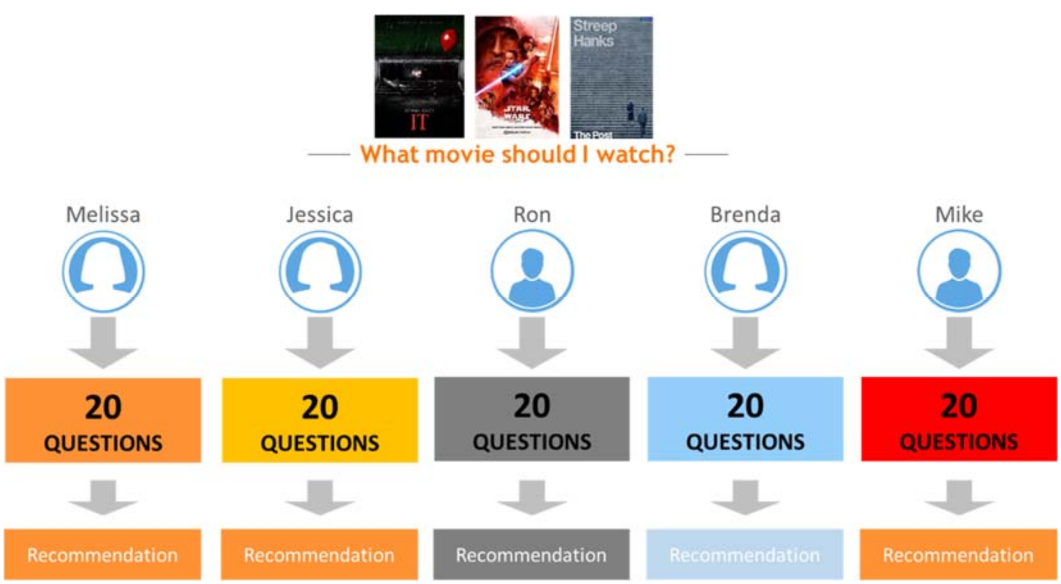
Suppose you're very indecisive, so whenever you want to watch a movie, you ask your friend Melissa if she thinks you'll like it. In order to answer, Melissa first needs to figure out what movies you like, so you give her a bunch of movies and tell her whether you liked each one or not (i.e., you give her a labeled training set). Then, when you ask her if she thinks you'll like movie X or not, she plays a 20 questions game with IMDB, asking questions such as "Is X a horror movie?" "Does Leonardo DiCaprio star in X?" and so on. She asks more informative questions first (i.e., she maximizes the information gain of each question), and gives you a yes/no answer at the end.

Thus, **Melissa is a decision tree for your movie preferences**.

But Melissa is only human, so she doesn't always generalize your preferences very well (i.e., she overfits). In order to get more accurate recommendations, you'd like to ask a bunch of your friends and watch movie X if most of them say they think you'll like it. That is, instead of asking only Melissa, you want to ask Ron, Jessica, Brenda, and Mike as well, and they vote on whether you'll like a movie (i.e., you **build an ensemble classifier**, aka a forest in this case).



Now you don't want each of your friends to do the same thing and give you the same answer, so you first give each of them slightly different data. After all, you're not absolutely sure of your preferences yourself – you told Melissa you loved *The Pursuit of Happyness*, but maybe you were just happy that day because it was your birthday, so maybe some of your friends shouldn't use the fact that you liked *The Pursuit of Happyness* in making their recommendations. Or maybe you told her that you liked *Alice In Wonderland*, but actually you really loved it, so some of your friends should give *Alice In Wonderland* more weight. So instead of giving your friends the same data you gave Melissa, you give them slightly different versions. You don't change your love/hate decisions; you just say you love/hate some movies a little more or less (formally, you give each of your friends a bootstrapped version of your original training data). For example, whereas you told Melissa that you liked *Inception* and *Harry Potter* and disliked *Avatar*, you tell Ron that you liked *Inception* so much you watched it twice, you disliked *Avatar*, and don't mention *Harry Potter* at all.



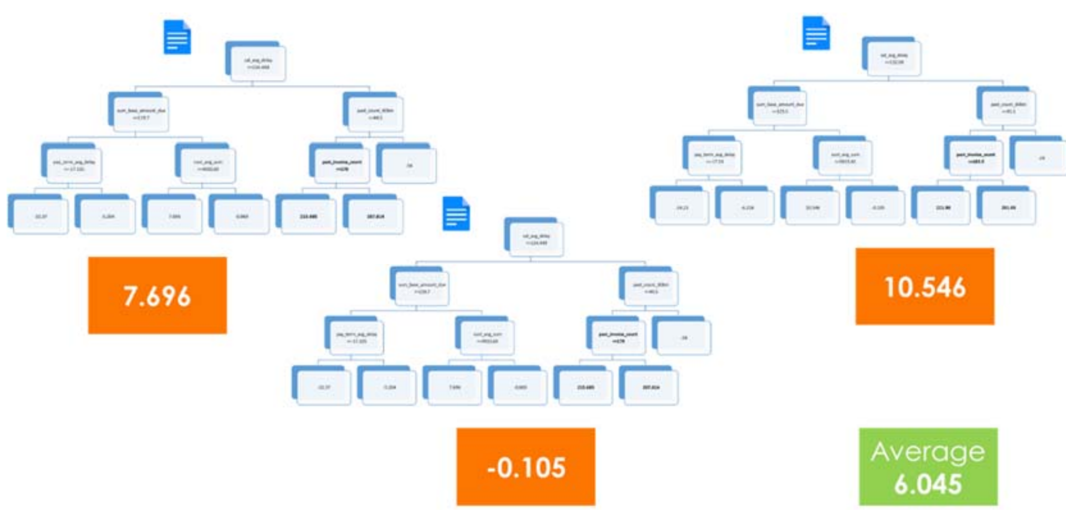
By using this ensemble, you hope that while each of your friends gives somewhat idiosyncratic recommendations (Melissa thinks you like vampire movies more than you do, Ron thinks you like Pixar movies, Brenda thinks you love watching fiction, and Mike thinks you just hate everything), the errors get canceled out in the majority. Thus, your friends now form a bagged (bootstrap aggregated) forest of your movie preferences.

There's still one problem with your data, however. While you loved both *The Pursuit of Happyness* and *Men In Black*, it wasn't because you like movies that star Will Smith. Maybe you liked both movies for other reasons. Thus, you don't want your friends to all base their recommendations on whether Will is in a movie or not. So, when each friend asks IMDB a question, only a random subset of the possible questions is allowed (i.e., when you're building a decision tree, at each node you use some randomness in selecting the attribute to split on, say by randomly selecting an attribute or by selecting an attribute from a random subset). This means your friends aren't allowed to ask whether Will Smith is in the movie whenever they want. So, whereas previously you injected randomness at the data level by altering your movie preferences slightly, now you're injecting randomness at the model level by making your friends ask different questions at different times.

And so, your friends now form a random forest.

Implementing the Random Forest Regression Model in A/R

As mentioned in the previous section, the collections process encompasses a sizeable number of factors (features) which lay the foundation of the regression analysis which in turn is leveraged for payment date prediction. The Random Forest Model is ideal for collections as it follows a simple rule: the larger the number of decision trees in the forest, the more accurate the result. These decision trees are inherently decision support tools which use tree-like graphs to show the possible consequences. If a training dataset with targets and features is input into the decision tree, it will formulate a set of rules. These rules are used to perform predictions. The ensuing figure illustrates three distinct decision trees which predict delay, based on unique combinations of different collections features:



The average delay in payment based on the three decision trees above comes out to be 6.045 days.

However, a random forest with three decision trees with a negligible number of features and combinations is grossly inadequate to predict payment trends for thousands of customers. The following figure shows a random forest regression model with a depth of 5.

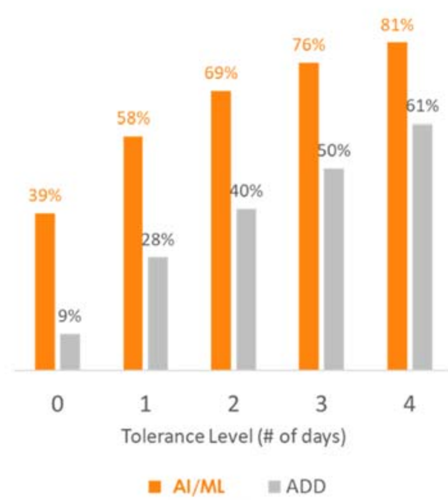
The subsequent figure shows the random forest regression model used to predict payment dates for collections:

Comparison with ADD

In today's aggressive economy, it is time for collections teams to recognize the necessity and worth of a proactive collections management process. This will only be achieved with collections strategies and correspondence game plans that resonate well with dynamic variables in the collections process. This chapter draws a comparison between the conventional payment estimation metric, ADD, and the AI/ML enabled collections management platform which operates on dynamic factors.

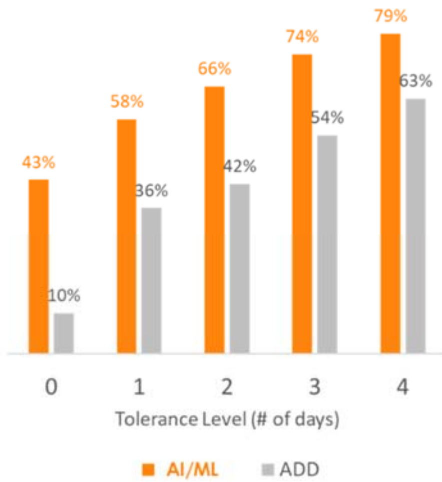
<u>For the entire dataset</u>

Tolerance	AI/ML	ADD
Zero Days	39%	9%
+/- 3 days	76%	50%



For the dataset of delayed invoices

Tolerance	AI/ML	ADD
Zero Days	43%	10%
+/- 3 days	74%	54%



In the above figures, tolerance refers to the difference between actual payment date and predicted payment date. This graph represents the performance of both AI/ML algorithm and ADD for payment date prediction for the entire dataset. Higher tolerance levels provide more scope of error in prediction accuracy. For instance, let's take the example of evaluating accuracy at a tolerance level of 7. This evaluation considers that if the predicted date for an invoice is within 7 days of the actual date, it should be labeled as accurate.

Therefore, the rising trend of accuracy for ADD at higher tolerance levels is not a real indicator of its performance. The AI/ML algorithm distinctively provides more accuracy than ADD for all tolerance levels. The table below summarizes the findings of this comparison.

Results Summary

	Accuracy	
	AI/ML	ADD
All Invoices	76%	50%
Delayed Invoices	74%	54%

AI/ML in Action

Leveraging AI/ML in Collections

Technology has been instrumental in revamping the collections process with centralized, real-time data access, automated customer collaboration and additional functionalities including logging communication and providing self-service portals to customers. However, when it comes to enabling key-decision making, the A/R teams were left on their own to deal with prioritizing accounts based on mountains of A/R information. **UNTIL NOW!** With Artificial Intelligence, technology has now forayed into a space where it is able to segregate accounts based on dynamic factors, drive resolution for low-value transactions for non-critical accounts, and simultaneously assist with decision making for the high-value, at-risk accounts.

Proactive Collections Management not only helps analysts de-clutter their collections worklists by prioritizing the right accounts, but also enables greater output with less cost by optimizing collections efforts. Insights from Artificial Intelligence could be ingrained in grassroot-level operations such as customer segmentation and collections rules and strategies.

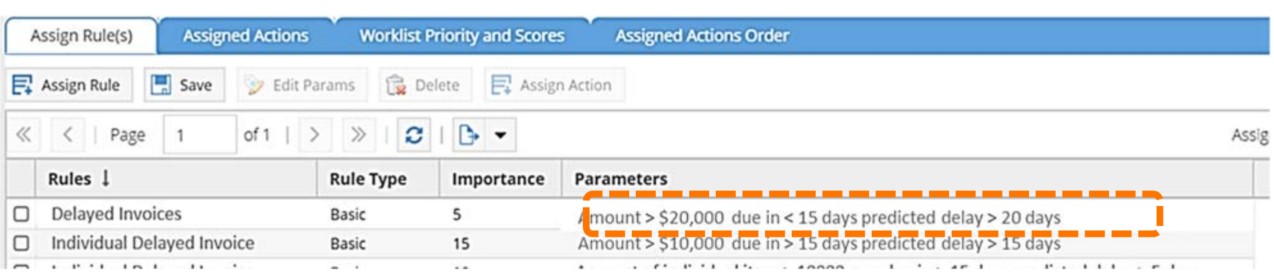
The collections teams gain three-fold support in enhancing their collections output:

1. **Risk Identification**: Identifying all potentially delinquent accounts through consideration of historical data and real-time credit analysis for each customer
2. **Worklist Prioritization**: Prioritizing worklists based on dynamic changes and insights driven by Artificial Intelligence and Machine Learning
3. **Collections Strategy**: Implementing best-practices across the industries combined with AI recommendations

Simulated Benefits

Proactive Collections Management is the ultimate goal for all A/R financials across the business spectrum. Improving customer satisfaction, ensuring better visibility, and optimizing resource allocation are some of the key benefits offered by an AI-enabled collections process. The following explores some of the simulated benefits in detail.

- **Dynamic Rules and Strategies**
  Predicted payment date and dynamic parameters enhance the collector's focus on critical accounts. For instance, the following figure illustrates a sample rule in a proactive collections management tool:



This rule is based on three factors:

- **Invoice value**: Static parameter from open A/R
- **Number of days for invoice to be due:** Dynamic parameter calculated from open A/R
- **Predicted delay**: Proactive parameter predicted by AI/ML algorithm

- **Optimize Order and Credit Management Based on Customer Behavior**
  Predicted Delay could be computed at the time of order creation based on the invoice parameters and customer history. If the predicted delay is high:
    - Request **upfront payment** for accounts or particular invoices
    - Require **payment commitments** at the time of order creation
    - Update credit terms to proactively **minimize delay** in payment

- **Proactive Correspondence to Reduce DSO and ADD**
  With proactive collections management, the collectors do not need to wait for invoices to be past-due and can take proactive actions based on predicted delay. Taking proactive measures reduces past-due A/R and ensures visibility into the performance of individual collectors as well as collections strategies.

Summary: Disruption for Collections?

The evolution of technology for collections has focused on how it has enabled easier access to data and more accurate and up-to-date aging reports for collectors.

In the 90s, collections teams were mainly leveraging custom programs written over mainframe or ERP databases. Aging reports were prepared on a daily or a weekly basis and would be generated on a printer. A team of collectors would then "dial for dollars" via phone calls.

The early 2000s saw the emergence of Data Warehousing and Business Intelligence tools such as Cognos, Business Objects, and Crystal Reports. These tools added the ability to slice-and-dice the data to the regular aging reports before contacting a customer. While this gave the teams more context to filter whom to contact and what invoices to push for, the analytics were still performed manually.

This decade has seen the emergence of software vendors with dedicated software products for Collections Management. Similar to Customer Relationship Management (CRM) tools for sales teams, the Collections Management software emerged as a standard part of collection departments. This third-generation technology brought fundamental automation via collections strategies and rules to completely eliminate the need for collectors to manually perform analytics to figure out which customers to contact. The system did this work for them. With the growing adoption of email, a good portion of phone calls started shifting to email communications instead, which in many standard cases such as past-dues or monthly statements, were automated via dunning and correspondence techniques.

Overall, third-generation collections technology enabled a 30-40% automation of collection operations and an average of 10% in DSO improvement – a definitive strategic impact to working capital.

This raises the question - what next? There is one flaw which still holds collections teams back. The entire collections process is still very reactive at its most basic function. A customer is contacted "after the fact" - when a customer is already delinquent on payments.

Artificial Intelligence and Machine Learning are already changing our lives. The Uber ride taken to work and the Amazon products bought online were all enabled by Artificial Intelligence. Apply this technology to create AI-powered Collections Management which continuously monitors customer data and predicts customer default well before an invoice is due to advance collections customer contact activity by five to fifteen days. In essence, change collections operations from reactive to proactive with an ability to further impact DSO and working capital by another 10% simply by empowering collections with Machine Learning.

Sonali Nanda is the Associate Vice President of Product Management at HighRadius, leading the development of the Integrated Receivables Cloud Platform and other product management initiatives, including the home-grown Rivana Machine Learning platform. She is responsible for building an effective product management team and defining product strategies and roadmaps, which accelerate sales and create happy customers.