IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Improving Collections Efficiency in Account Receivables systems using Predictive Analytics

Abdul Jabbar, Principal Cloud Architect, Hyderabad, India.

Abstract: Account receivables (AR) systems are a critical part of financial system of any business. Effective management of AR and financial performance of firms are positively correlated. Invoice to Cash (I2C) process, is a critical part of AR, which starts from the moment an invoice is created until the moment the customer's debt (payment) is settled or reconciled. Inefficient AR management delays cash realization, causing financial crunch of a given firm.

In this context there have been attempts to address the issue of outstanding receivables through improvements in the collections strategy, specifically, through supervised learning to build models for predicting the payment outcomes of newly-created invoices, thus enabling customized collection actions tailored for each invoice or customer. Such models can predict if an invoice will be paid on time or not and can provide estimates of the magnitude of the delay. Predictive models use analytics to predict probability of event to happen. To effectively predict payment behavior of customers, the data that is generated in AR system has to be accurate in quality and interpretation, if not, may result in wrong predictions.

Current predictive models which are based on business rules need to be data driven and the data has to be accurate, usage of proper algorithms, multiple models to make predictive models more comprehensive and thereby resulting in better business decisions. Improving approach of predictive models and data accuracy feeding into Predictive Model for Invoice scoring and usage of appropriate algorithms and comprehensive approach are dealt in this paper. The algorithm being proposed gives a statistical model, which uses business domain based variables

Keywords - Account receivables, Open Items, Closed Items, Touch Data, Days Past Due (DPD), Payment Term, Self-Cure, Ageing Bin, and Segmentation

I. Introduction

The Invoice-to-Cash (I2C) process describes a composite business process that comprises the necessary steps to fulfill an order for a good or service, from order entry to payment receipt. It involves the process from the moment the invoice is created until the moment the customer's debt (payment) is settled/reconciled. Invoice to Cash process involves the process from the moment the invoice is created until the moment the customer's debt (payment) is settled/reconciled.

While the number and nature of such steps may vary depending on the type and size of the firm, most I2C processes follow a similar high level workflow. Overall process can be summarized as a set of four sub processes, such as Customer Billing or Invoicing, which involves raising bills to customer, which can be performed in different ways; such as manual entry, created based on Shipment Details, created using information from a Sales Order or information available on existing invoice can be copied the same Issuing the invoice. Secondly, Making Adjustments on Invoices or Outstanding Dues. There could be various types of adjustments that might be required once the invoice has been raised on the customer. Some examples are; Adjustments; for various reasons like goods returned, shipping delays, tax differences etc., Disputes; The invoices raised can be disputed by the customer, which needs to be tracked and appropriately closed. Third sub process would be Receiving and Managing Payments, which can be done in the form of Cash, Cheque, Bank Transfer etc. Fourth and final sub process Application and Accounting of the Cash Received.

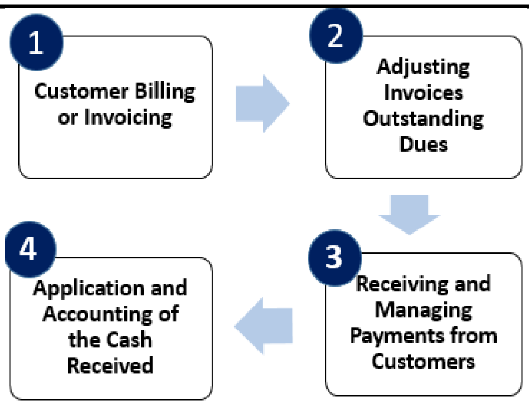
1
Customer Billing or Invoicing
2
Adjusting Invoices Outstanding Dues
3
Receiving and Managing Payments from Customers
4
Application and Accounting of the Cash Received

Fig 1: Invoice to Cash (I2C) Process Flow

In this paper we concentrate on the step 3 of the account receivables process, which is related to collections process. It is the core of the AR system and deals with tracking open invoices, account profiling, contacting customers, tracking the invoice status to closure. Most often, these steps are generally manual and hence, slow, expensive, and inaccurate, despite their importance to the business.

Efforts have been made to make this process predictive and feed artificial intelligence through Machine Learning. However the machine learning models that have been evolved for solving this have been very basic, which are founded on business rules and less on data analysis. Existing Predictive models are having some inherent weakness, as mentioned under:

- Based probability that the invoice will be paid on time.
- Missing customer, collector, or client behavior input in defining strategy
- Customer Contact Strategy relies on volume dollar thresholds to set initial systemic collector activity
- Threshold set is based on volume of activity it drives in relation to the capacity of the organization.
- Lack of data visibility drives subjective approach as one global solution vs a segmented approach.

Current predictive models need improvement from data accuracy, usage of proper algorithms, multiple models to make predictive models more comprehensive and thereby resulting in better business decisions. Also the data to be fed into such a model has to be more frequent and accurate. Such inherent weaknesses in existing data models have resulted in working on areas of improvement and coming up with new predictive data models for collection activity based on improving data accuracy through a variety of means.

Improving data accuracy feeding into Predictive Model for Invoice scoring and usage of appropriate algorithms and comprehensive approaches are proposed in this paper

II. Payment Prediction of Invoice

A. Terminologies

Below are explanation of terminologies related to invoice processing, which are used frequently in the paper:

- Invoice to Cash: It is a process involves the process from the moment the invoice is created until the moment the customer's debt (payment) is settled/reconciled. Typical I2C process constitutes the following actions
- Closed Items : Closed invoices data, contains information about customer, invoice number, invoice release date, payment date, due date, and amount receivable from the customer
- Open items : Contains same information as closed items but these invoices are outstanding
- Touch Data : Contains information about collection efforts made for a particular invoice like invoice number, contact date/time, type of contact made, and outcome of the contact
- Payment Data : Contains payment information like amount paid ,mode of payment, and payment date against particular invoice in close data
- Days Past Due(DPD) : # days after due the payment was received
- Payment Term : Difference between due and invoice date for a particular invoice
- ILP: Invoice level Priortization program, Scans invoices to prioritize them for collectors to act upon. Scheduling logic is there for collectors to act upon
- Collector: An individual who tracks and monitors all the invoices to closure.

B. Current Model vs Proposed Models

Current models are based on business metrics such as DSO (Days Sales Outsourcing) and are restricted in terms of frequency of invoice loads that happen on a weekly basis and are typically remotely hosted when compared to Accounts Receivable system. Also the data feeds happened

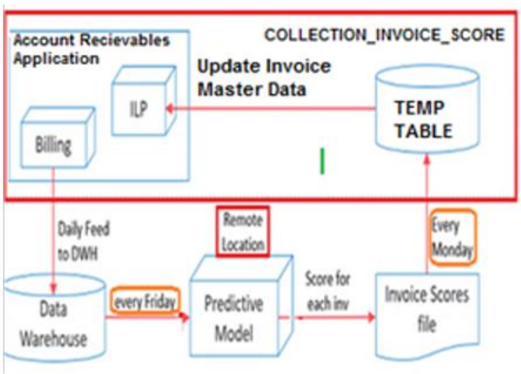

Fig 2. Current Predictive Model Architecture

Following are proposed as an alternate

- Invoices to be fed into model on a daily basis and scored on a daily basis.
- Retraining model with historical data every 3 months.
- The model should use invoice score based on both new and historical information to estimate the probability of the invoice to be paid on time. This score will be provided back to Account Receivables system, which will be read by the ILP process so that invoices with high probability to be paid on time will be excluded from the Collectors' Due Task List
- The data feed using the file feed approach should be replaced with direct database access using a separate schema where data for the model will be provided daily
- Re-scoring for previously scored open invoices to happen on Adhoc basis. It will be requested and once requested, this will be applied to all open invoices. By default, this functionality will be off.

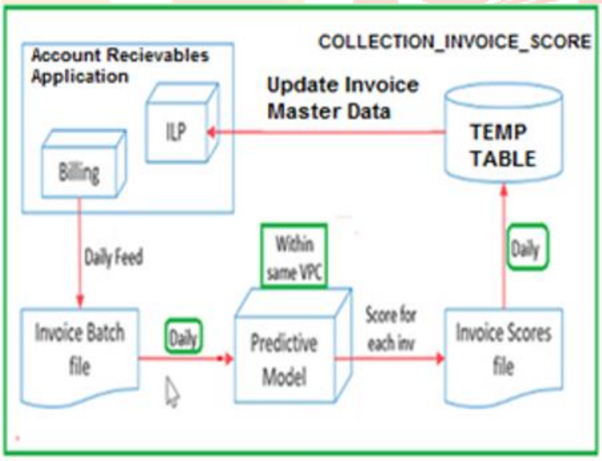

Fig 3. Proposed Predictive Model Architecture

In the light of above below models are proposed

- Self-Cure Model : Build a predictive model to predict if an invoice will be paid on time with what's the probability an invoice will paid on time
- Ageing Bins Model : Build a classification model to predict by how many days an invoice may be delayed in case the invoice is found out be paid late
- Customer Segmentation : Segment the customer portfolio from collections perspective using transactional data, collection efforts activity data, and payment data, and identify group of customers who exhibit similar behavior

C. Processing overview

Data needed for overall processing, should focus on Data Preparation, Feature Engineering and Feature Selection, which have below tasks

- Initial Data Preparation: Develop an exhaustive data dictionary by analyzing all the fields from invoices, fields from touch data and closed data
- Variable Transformation: Date columns treated as string fields 2 numerical field treatment
- Feature Engineering comprises features using Customer data, touch data, invoice level features.

- Feature Selection encompasses Numerical Feature Selection developed using correlation matrix to eliminate and select the features, and Categorical Feature Selection through Chi-Squared tests performed to check dependency between categorical variables. As a result develop univariate logistic regression models to select the best features for modelling
- Post Modeling- Feature Selection: Eliminated features interpreting the p-values , and using VIF after building the model

D. Evaluation Strategies

Predictive model propose will be subjected to below validation tests

- Validation on closed items – the model will be tested against data related to two consecutive quarters of a given financial year.
- The setup validation will be mimicked by validating the model against the invoices that are billed on specific day and got the propensity to pay. Generated the accuracy views of every day for 3 consecutive months. Validated the model accuracy using K-fold cross validation and Time series cross validation techniques
- Deliver customer segmentation model and use it with conjunction with self-cure model
- Build and deploy the ageing model to production

III. APPROACH AND ARCHITECTURE

A. Proposed Approach

Predicted score is the model score of 0 or 1 that will be fed to ILP program. Where 0 means will likely pay on time; and 1 means will NOT likely pay on time. Refine the feed to pass only the records with Closed status instead of just checking Closed date as there could re-opened invoices with closed date Parameterize the date to be used as start date for processing and checking invoice input date. Update all the database links since the model will be migrating to backfeed database to save on cost. Update the appropriate field in Invoice Extension table which ILP program reads

B. Data Flow

Predictive model would have direct access to the AR system's database as it will be located as one of its schema. Also it is proposed to have a staging table as a staging area to store validated data before being loaded into actual table. Below is the proposed data flow diagram for the proposed

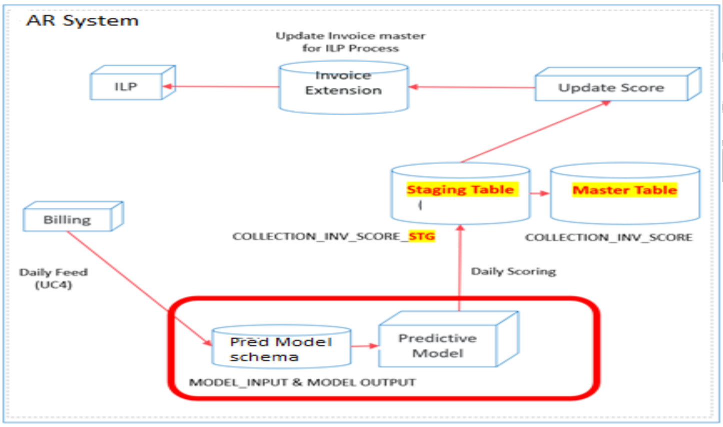

Fig 4 : Data Flow diagram of the proposed Predictive Model

New Staging table will be created so that AR system data for processing is not exposed to another or external process. The new Staging (Stg) Table is proposed to have columns like INVOICEID; predicted score; propensity to pay; actual score; score date; timestamp; USERID (PROCESSID); model name. Unique keys is the combination of INVOICEID and Score Date and USERID.

Predicted score is the model score of 0 or 1 that will be fed to ILP process. Where 0 means will likely pay on time; and 1 means will NOT likely pay on time. Propensity to pay will provide the probability that the invoice will be paid on time. This will have a value between 1 or 0 similar to % probability. This is the one generated before the predicted score is provided. Actual score will be calculated by the model team based on the actual closed date for Closed Invoices during the retraining.

USERID will capture the processor/system id. A new USERID or PROCESSID will be added if a new Predictive Model or processor is used to feed the score i.e., to differentiate a Processor (Model for business A) from Processor (Model for business B). This will be used for system access control and should not be confused with business data access control, unless required for the processor to have multiple database accounts basis business data.

New MASTER table is proposed to capture the information from STG table plus the Process Flag. It will have the same structure as the STG table. But unlike the STG table, MASTER table will not be exposed to other processes. There will be a trigger to populate the Master table (in case of insert/update). The Stg table will be purged weekly as long as the records already exist in Master table. The Master table will be purged based on Retention Policy

C. Data Elements

Data is analyzed basis feedback received from the collections teams. Below invoice related details are proposed to be captured so that the invoices can be tracked effectively.

FIELD	DESCRIPTION
BUSINESS_ID	Business ID
BUSINESS_NAME	Business Name
CC_POLE	C&C or Collector Pole
CLIENT_BILLING_DOC_NO	Invoice Number or Client Billing Document Number
CLIENT_DOCUMENT_DATE	Invoice Date
CLOSE_DATE	Invoice Close Date
COLLECTOR_CODE	Credit and Collections Number
CONV_OUTSTANDING_AMT	Outstanding Amount in USD
CONVERTED_ORIGINAL_AMT	Invoice USD Converted Original Translated Amount
CUST_SERVICING_TYPE	Servicing Type
DEBTOR_NAME	Customer/Debtor Name
DEBTOR_NO	Customer/Debtor Number
DISPUTED	Dispute Flag – Will be set to “Y” if invoice is disputed, else “N”
DISPUTE_CODE	Dispute Code
DISPUTE_CODE_DESCRIPTION	Dispute Code Description
DISPUTE_COMMENT	Dispute Comment
DISPUTE_DATE	Dispute Date
DISPUTE_NUMBER	Dispute Number
DISPUTE_RESOLUTION_DATE	Dispute Close Date
DISPUTE_STAGE	Dispute Stage
DISPUTE_STATUS	Dispute Status
DOC_TYPE_CODE	Transaction / Doc Type Code
DOC_TYPE_DESCRIPTION	Transaction / Doc Type Code Description
DOCUMENT_STATUS	Invoice Status
DUE_DATE	Invoice Due Date
DYNAMIC_DPD_ANALYSIS	Days Past Due (DPD) equals to Invoice Close Date minus Invoice Due Date
FUNCTIONAL_CURRENCY	Functional Currency
IC_NAME	Investment Code Name
IC_NUMBER	Investment Code Number
IC_POLE	Client Entity or Bus. Pole
INVOICE_ID	Unique Invoice ID
INVOICE_TYPE	Invoice Type
INVOICE_INPUT_DATE	Date when invoice is loaded
INV_SYSTEM_DATE	Date when invoice is loaded and released from Billing Error if there is
LAST_UPDATED_DATETIME	Date when data was sent (last sent) to the Model
PARENT_CUSTOMER_NUMBER	Parent Customer/Debtor Number
TRANSLATED_ORIGINAL_AMT	Invoice Original Translated Amount

Table 1.1: Invoice related data elements and its descriptions.

D. Batch Processing

Batch job are proposed to be scheduled to populate the table which the model will read for scoring the invoice. This job will have to run daily. A separate Batch job would provide the Invoice score back to ILP. This job will run daily.

E. Additional Optimizations

Preprocessing the feed:

- Refine the feed to pass only the records with Closed status instead of just checking Closed date as there could re-opened invoices with closed date
- Parameterize the date to be used as start date for processing and checking invoice input date
- Database links have to be updated if the Model is migrated to AR system's backfeed database to save on cost

F. Implementation approach

Below is a high level implementation approach

- Required infrastructure for setting up Predictive Model setup is proposed to be done in any of the cloud IaaS and install the required software and libraries. As this model needs a database schema, it is proposed to have a separate database instance for the model or accommodate in existing database of AR system.
- Initial data preparation, Feature engineering and feature selection, evaluation strategies have to be performed as mentioned in the previous sections.
- Next step is to build the Model and create the required interfaces that involve, creating a database view against invoice table to select invoices with no scoring, building program to pull invoices for scoring purposes, updating "COLLECTION_INV_SCORE" table in backfeed database and reviewing table structure that will be updated by model
- Once the model is setup, next step would be to retrain the Model, by getting the updated scores/logic back, Strategy Set Up, creating or amending strategies for Equip customers and service customers, link strategies to C&C's for equipment/services split, set up rules & tasks under each segment/bucket, followed by thorough testing of the overall system. Post deployment changes can be made to strategy set ups.

IV. RELATED WORK

- There are a number of vendors offering pre-packaged solutions for Invoice to Cash (I2C). Examples are Oracle's EBS Special Edition Order Management and SAP's Order-to-Cash Management for Wholesale Distribution. Oracle's solution provides information visibility and reporting capabilities. SAP's solution supports collections and customer relationship management. As on date, none of such solutions incorporates analytics, specifically predictive modeling for improved prioritization of invoices or for customer ranking, with subsequence collection process optimization.

- Predictive modeling approaches are widely used in a number of related domains, such as IT infrastructure management and ticketing systems. Amazon Web services proposes a machine learning based model to predict failure of EC2 instance (VMs on AWS) through a machine learning model that detects potential equipment failures and provides recommended actions to take. CloudFormation template that deploys an example dataset of a turbofan degradation simulation contained in an (Amazon S3) bucket and an Amazon SageMaker endpoint with an ML model that will be trained on the dataset to predict remaining useful life (RUL). SageMaker notebook instance orchestrates the model, and another SageMaker training instance performs the training. The training code and trained model are stored in the solution's Amazon S3 bucket. Amazon CloudWatch Events rule is required that is configured to run once per day, to trigger serverless AWS Lambda function that creates an Amazon SageMaker batch transform job that uses the trained model to predict RUL from the example dataset.

V. CONCLUSION

In this paper, we have presented a supervised learning approach and the corresponding results in the context of AR collections. We developed a set of aggregated features which capture historical payment behavior for each customer. Our results show that having this set of features enhances the prediction accuracy significantly, and it is more valuable in predicting payment delays for invoices from returning customers than using customer features. However, we also observed that customer features play an important role in the prediction of payment delays when no historical information is available, e.g., for invoices from first time customers. We demonstrate that by using cost-sensitive learning, we are able to improve prediction accuracy particularly for high risk invoices, which are under-presented in the data sets.

REFERENCES

1. Predictive Modeling for Collections of Accounts Receivable by Sai Zeng, Prem Melville, Christian A. Lang, Ioana Boier-Martin, Conrad Murphy. 2007
   https://dl.acm.org/doi/10.1145/1288552.1288558
2. Customer Level Predictive Modeling for Accounts Receivable to Reduce Intervention Actions, Michelle LF CHEONG, and Wen SHI, August
   https://csce.ucmss.com/cr/books/2018/LFS/CSREA2018/ICD8000.pdf
3. How a scientific approach to accounts receivable can give you more of the working capital you need
   https://www.genpact.com/insight/solution-overview/how-a-scientific-approach-to-accounts-receivable-can-give-you-more-of-the-working-capital-you-need
4. Recievable Analytics
   https://www.capgemini.com/wp-content/uploads/2017/07/receivable_analytics.pdf