

A PROJECT REPORT

on

**AI-Enabled Fintech B2B Invoice Management
Application**

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

**BACHELOR'S DEGREE IN
INFORMATION TECHNOLOGY**

BY

SOUHARDYA PAL

1906511

UNDER THE GUIDANCE OF

**Sayak Mondal, Pushpita Saha, Nishi Anand, Varun Jha, DuuviKalyanRam
HighRadius, Bhubaneswar**



SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

May 2022

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024



CERTIFICATE

This is certify that the project entitled

AI-Enabled Fintech B2B Invoice Management Application

submitted by

SOUHARDYA PAL 1906511

is a record of bonafide work carried out by HIM, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2022-2023, under our guidance.

Date: 15/04/2022

Sayak Mondal, Pushpita Saha,
Nishi Anand, Varun Jha,
DuuviKalyanRam

Project Mentor

Acknowledgements

I am profoundly grateful to **Sayak Mondal, Pushpita Saha, Nishi Anand, Varun Jha, DuuviKalyanRam** of **HighRadius, Bhubaneswar** for their expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. I would like to express my deepest gratitude for all the assistance and support without which this project would not have been able to reach its fulfillment. I have taken efforts in this project. However, it would not have been possible without the generous encouragement and help of many individuals with their abilities. I am also thankful to my POD leaders and moderators who guided me throughout the course of the training with their valuable inputs, cooperation and guidance.

I am also thankful to Kalinga Institute of Industrial Technology, Bhubaneswar for providing me technical skills and facilities which proved to be very useful for our project.

SOUHARDYA PAL
1906511 (IT)

ABSTRACT

An invoice management system is any tool or platform that tracks sales, orders, inventory, and fulfillment as well as enables the people, processes, and partnerships necessary for products to find their way to the customers who bought them.

In the B2B world, Account Receivables play a very major role for businesses to run smoothly. The aim of this project was to develop a Sales Management Application which would be artificially enabled. The Sales Management Application has an Account Receivables Dashboard which can predict invoice payment dates and categorize them into 4 different aging buckets based on payment date/delay. The ML dataset is based on a regression model which uses different regression algorithms to predict the delay in payments of invoice. This B2B application helps the users to understand their clients better on the basis of aging buckets, delay the customer's make and act accordingly.

One of the biggest problems that businesses face today is poor payment practices where their customers fail to make payment on time. This has a very large impact on businesses especially SMEs who face the problem of having a limited cash flow. Late payments problem has resulted in businesses facing severe consequences and sometimes, causing them to shut down their business. In fact, many businesses have to resort to tools like invoice finance to help overcome the problems of receiving late payment.

Its responsibilities include:

- Collecting payments from customers for their past due invoices
- Sending reminders and follow ups to the customers for payments to be made
- Looking after the entire process of getting the cash inflow
- Help the company get paid for the services and products supplied

As a result, this project creates a user-friendly application with the goal of providing correct output results even when a big number of customers are involved. A diverse end-to-end application has been built using complex technologies and an elaborate framework that stands out in the market owing to its usage of new technologies such as Machine Learning, Artificial Intelligence, React, and others. The application consists of two parts: a comprehensive Data Science and Machine Learning model and an application design and user interface development methodology.

Keywords: Machine Learning, MySQL, HTML & CSS, JavaScript, Java Servlets, Flask Framework, ReactJS

Contents

1	Introduction	1
2	Basic Concepts	2-4
3	Problem Statement / Requirement Specifications	5-9
4	Implementation	10-15
5	Conclusion and Future Scope	16
	6.1 Conclusion	16
	6.2 Future Scope	16
	References	17
	Plagiarism Report	18

List of Figures

1	Screenshots of ML	11
2	Screenshots of SQL	12
3	Screenshots of UI	12-15

Chapter 1

Introduction

The B2B operated differently from B2C or C2C world. A computer software package that allows firms to manage their orders and inventory is known as an invoice management system. Invoice management systems aid in more accurate inventory management by automatically entering new inventory into the system, tracking sales across various selling platforms such as eBay and Amazon, and alerting you, the business owner, when your stock of a particular item drops low enough to facilitate a reorder.

Beginning with the client order and continuing through payment reconciliation, fulfilment, and shipment, an invoice management system can automate the order-to-cash process. Invoice management software can be used by organisations of all sizes, both B2B and B2C.

Effective bill management improves the business work-flow and increases the likelihood of repeat customers.

Chapter 2

Basic Concepts

The project was broadly divided into two parts:

1. Machine Learning:

To build a Machine Learning model to predict the date of payment of an invoice.

2. Java and React:

The backend and the frontend, which is to build an AI-Enabled FinTech B2B Invoice Management Application which would also integrate the Machine Learning model, to predict the delay of invoices

2.1 MACHINE LEARNING

The objective was to build a Machine Learning Model to predict the payment date of an invoice when it gets created in the system. It would also categorize the invoice into different buckets based on the predicted payment date. Clicking on the Predict button will populate the Predicted Payment Date and the Predicted Aging Bucket column of the grid with the predicted values derived from the ML model. The different buckets would be 0- 15 days, 16-30 days, 31-45 days, 46-60 days and greater than 60 days.

2.1.1. The order is placed

Your consumer places an order using a third-party sales site, your own website, or a real agent on the phone. Customers will enter their information on a standardised form online, with the option of saving a secure preferred payment method. Make all fields of your online form required to optimise the sales process and ensure you get all of the relevant contact information for the customer up front. This establishes a customer profile, which your invoicing management system may use to keep track of their purchase history, order volume, and payment and delivery preferences. It also provides their phone number and email address in case you need to contact them regarding service recovery. Once your software system accepts the charges, the payment is executed, and the order is sent to the warehouse

2.1.2. Warehouse processing

Once the order arrives at the warehouse, it's checked by the intake team and the item or items are "picked" from the stock. Having a SKU and barcode for every item increases the accuracy of fulfillment and makes it easier for pickers to simply scan the item and add it to the order. If there isn't enough of the item(s) in stock to fulfill an order, then a purchase order is automatically placed through the invoice management software. You and the warehouse manager will receive an alert that there may be a delay in fulfillment. Your customer may receive an automatic notification of the delay, and the customer service team can follow up with your customer.

2.1.3. Reconciling the order

Next, the order is sent to the accounting department or preferably it should sync automatically with your cloud accounting software, where it's recorded in your A/R ledger. The sale is logged and a receipt sent to your client. Automating your sales ledgers makes it easier for auditing, inventory reconciliation, and end-of-year taxes.

2.1.4. Shipping the order

Once the order is picked from the warehouse, your packing team will double-check for accuracy, again using the barcodes and SKU. Then, the order is packed carefully and shipped via a third-party delivery system. Your customer will receive a notice through the invoice management system that their order has shipped, along with a tracking number and estimated delivery time. As a store owner, you can also track the progress of shipped orders, which can be helpful if there are special needs orders, such as re-deliveries, VIP orders, or unusually large ones.

2.1.5. Post-sales follow-up

Once the order arrives, the software should generate an automatic email to follow up, asking how they liked the items and ensuring that they received everything accurately. This email should include detailed instructions on how to reach customer service if there are any issues, taking the frustration out of guessing how to obtain a refund if needed. Your customer service team oversees this process, thanking the customer for their business or working with them for a refund or replacement.

2.1.6. Special order oversight

Another aspect of good OMS is the ability to flag a special order. This may be a return replacement or it could be a VIP order that includes a free thank-you gift or special coupon. When these orders are placed through the system, the software can flag them with a code, allowing you or your customer retention team to personally monitor the order for accuracy.

2.2.2 FRONTEND

2.4.2. ReactJs

ReactJs is an open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality. In the project ReactJs is the frontend technology used. It is used to design the User Interface which is the dashboard of the Invoice Management Application.

The app was powerful and included several concepts such as working with concepts, hooks, redux and more, including endless scrolling, which is a web design that loads continuous content as a user scrolls down the page, eliminating the need for pagination. Redux is a state management tool. With redux the state of the application is kept in a store, and each component can access any state that it needs from this store. It allows React components to read data from a Redux Store, and dispatch Actions to the Store to update data. Redux helps apps to scale by providing a sensible way to manage state through a unidirectional data flow model.

2.4.3. Java Servlets

A Java Servlet is a Java software component that enhances a server's functionality. Although servlets can reply to a wide range of requests, they are most typically used to host web applications on web servers, making them a server-side servlet web API. Other dynamic online content platforms, such as PHP and ASP.NET, have Java counterparts in the form of web servlets. Java Servlets are utilised in the project to create an Application Programming Interface (API) that connects the backend and frontend properly.

Chapter 3

Problem Statement / Requirement Specifications

3.1 Problem Statement for MLModel

The objective of the first half of the winter internship project is:

- To build a Machine Learning Model to predict the payment date of an invoice when it gets created in the system.
- Categorize the invoice into different buckets based on predicted payment date.

We received an invoice dataset that contained the past payment information and behaviour of various buyers. Based on the previous payment patterns, the ML model needs to predict what will be the date a payment is made by the customer for an invoice.

The model also needs to predict which aging bucket the invoice falls into based on the predicted payment date.

3.2 High Level Requirements Of Application:

Specifically, below are the major aspects of the application that needs to be developed. The details for each of the below is provided in the functional overview section.

1. Invoices Dataset:

- a. HighRadius will provide you a invoices dataset which you need to parse and process.

2. AI Support in the application:

- a. Add support for predicting the payment date for invoice(s).
- b. UI should have a button to trigger the prediction of payment date.
- c. Payment date needs to be persisted across sessions in the UI.

3.3 Problem Statement for Application Development:

The objective of the second half of the winter internship project is:

- To build a full stack Invoice Management Application using ReactJs, JDBC, Javaand JSP.
- Build a responsive Receivables Dashboard.
- Visualize Data in the form of grids.
- Perform Searching operations on the invoices.
- Edit data in the editable fields of the grid.
- Download data of selected rows in predefined templates.

3.4 High Level Requirements Of Application

Specifically, below are the major aspects of the application that needs to be developed.

1. Data Loading in DB:

- a. You will be provided with a invoices dataset which you need to parse, process and load in the provided database schemas.

2. UI Representation of the data:

- a. Build a responsive UI which can display the invoice data loaded from the database.
- b. The UI should support searching and infinite scrolling operations.
- c. The UI should support editing of some editable fields, adding a new row to the grid, deleting rows from the grid and downloading selected records from the grid in predefined template(s).

3.5 The dataset that used had following features:

- i) business_code - company code of the account
- ii) cust_number - customer number given to all the customers of the Account
- iii) name_customer - name of the customer.
- iv) cust_number - Each customer has a number that uniquely identifies it.

- v) document_create_date - The date on which the invoice document was created
- vi) document_create_date_norm - Normalised date of the invoice document
- vii) posting_id - key indicator to identify whether an AR item is an invoice.
- viii) due_in_date - The date on which the customer is expected to clear an invoice
- ix) invoice_id - Unique number assigned when a seller creates an Invoice.
- x) baseline_create_date - The date on which the Invoice was created.
- xi) total_open_amount - The amount that is yet to be paid for that invoice
- xii) invoice_amount - The total amount for that invoice.
- xiii) cust_payment_terms -Business terms and agreements between customers and accounts on discounts and days of payment
- xiv) area_business - Business area in sap is defined as an organisational area within the financial accounting module.
- xv) clear_date - The date on which the customer clears an invoice, or in simple terms, they make the full payment.
- xvi) is_open_invoice - indicator of whether an invoice is open or closed.
- xvii) invoice_currency - The currency of the invoice amount in the document for the invoice.
- xviii) doc_id - It is also an unique identifier of an invoice and is a primary key.
- xix) total_open_amount - open amount of an invoice

3.6. Machine Learning Model

While building this model we had gone through many process:

- Step 1. Firstly we needed to import the dataset and also import various data manipulation libraries.
- Step 2 . We visualized the dataset in the form of a data frame to get a brief idea about the dataset.
- Step 3. Identified the target variable which is clear_date.
- Step 4. Independent and dependent variables identification and extraction.
- Step 5. Handling Missing Values using Null Imputation Techniques.
- Step 6. Encoding Categorical Variables to training purposes.
- Step 7. Splitting the dataset for train/test and validation.
- Step 8. Feature Scaling for improved training using Normalization and Standardisation.
- Step 9. Training and Validating ML model.
- Step 10. Predict clear_date using test data.
- Step 11. Prepare aging bucket by subtracting invoice creation date from predicted clear date.

The different buckets were :

- 1. 0-15 days
- 2. 16-30 days
- 3. 31-45 days
- 4. 46-60 days
- 5. Greater than 60 days

3.7. Objective of Java :

3.7.1. Data Loading in the Database:

Step 1: Execute the SQL script for the creation of table.

Step 2: Read the csv datasheet using a CSV reader and stored information.

Step 3: We used a JDBC driver and also made a POJO class which helped us load the datasheet into the database in batches.

3.7.2. Servlet Creation:

So after the UI is made some actions (add, edit ,delete etc.) needs to perform. So using the help of servlets given below we can make those actions happen.

1. Add servlet - Get a POST request from the frontend with parameters such as invoice amount, notes, date, etc and pass them to the SQL database.
2. Edit Servlet - GET a POST request from the frontend with parameters such as doc_id to identify the invoice in addition to the parameters which need to be changed.
3. Delete Servlet - Delete the selected invoices from the database by passing their respective doc_id's to identify them in the database.
4. Search Servlet - Get the invoice number from the frontend and pass them as a http request using axios to the backend and search through the database and return it to the frontend again.
5. Data Display Servlet - Display the table of invoices to the frontend.

3.8. Objective Of React (UI Representation)

3.8.1. Header Section

The header consists of :

1. Account name logo on the left
2. The HighRadius Logo in the center.

3.8.2. Grid Panel Section

The Grid panel section will be divided into 4 portions:

1. The header of the grid will have a Predict button on the top left corner followed by a View Correspondence Button, an Add Button, an Edit Button, a Delete Button and a Search Bar.
2. The name of the grid that is Invoice List will be mentioned in the top left corner of the grid.
3. The second portion is the table with customer invoice data as rows and the columns

List of all the columns to be represented on the UI are as follows:

1. Checkbox
2. Customer Name
3. Customer Number (Customer #)
4. Invoice Number (Invoice #)
5. Invoice Amount
6. Due Date
7. Predicted Payment Date
8. Predicted Aging Bucket
9. Notes

AI Enabled Fintech B2B Invoice Management Application

The following is the snapshot of the UI representation of the data or the frontend of the Invoice Management Application:

List of all the buttons present in the Grid Panel with their functionalities are as follows:

1. **Predict Button** - The Predict button will remain in disabled state if no rows are selected. Whenever one or more rows are selected, the Predict button will be activated. After clicking on the Predict button, the Predicted Payment Date and Predicted Aging Bucket will be populated for the respective records.
2. **View Correspondence Button** - Helps in viewing an invoice or a group of invoices in detail and exporting it.
3. **Add Button** - Adds an Invoice to the existing database
4. **Edit Button** - Edits an invoice in the database
5. **Delete Button** - Deletes an invoice in the database
6. **Search For Invoices** - Search an invoice by the invoice number

Chapter 4

4.1. Results Of The Project:

On our data set, we examined how the various features interact to produce findings that allow us to draw a number of inferences. Area business is an empty feature in the ML model and does not support the modelling process. Both doc_id and Invoice_id are exact duplicates of one another. Subtracting clear_date from due_in_date yields the delay date. Servlets facilitate the connection between our database and our user interface. The servlets created were for data retrieval, record addition, record editing, record deletion, record searching, and correspondence viewing in various templates.

4.1.1 Credit Management

Credit management refers to the process of giving credit, determining the terms on which it is issued, reclaiming the credit when it is due, and ensuring compliance with the company's credit policy, among other things.

In basic terms, the majority of B2B transactions are made on credit. The client permits the consumer to buy goods and services from it on credit at this step. The client informs the customer of all credit policies in place. In addition, the customer employs a variety of measures to optimise cash flow and lower DSO.

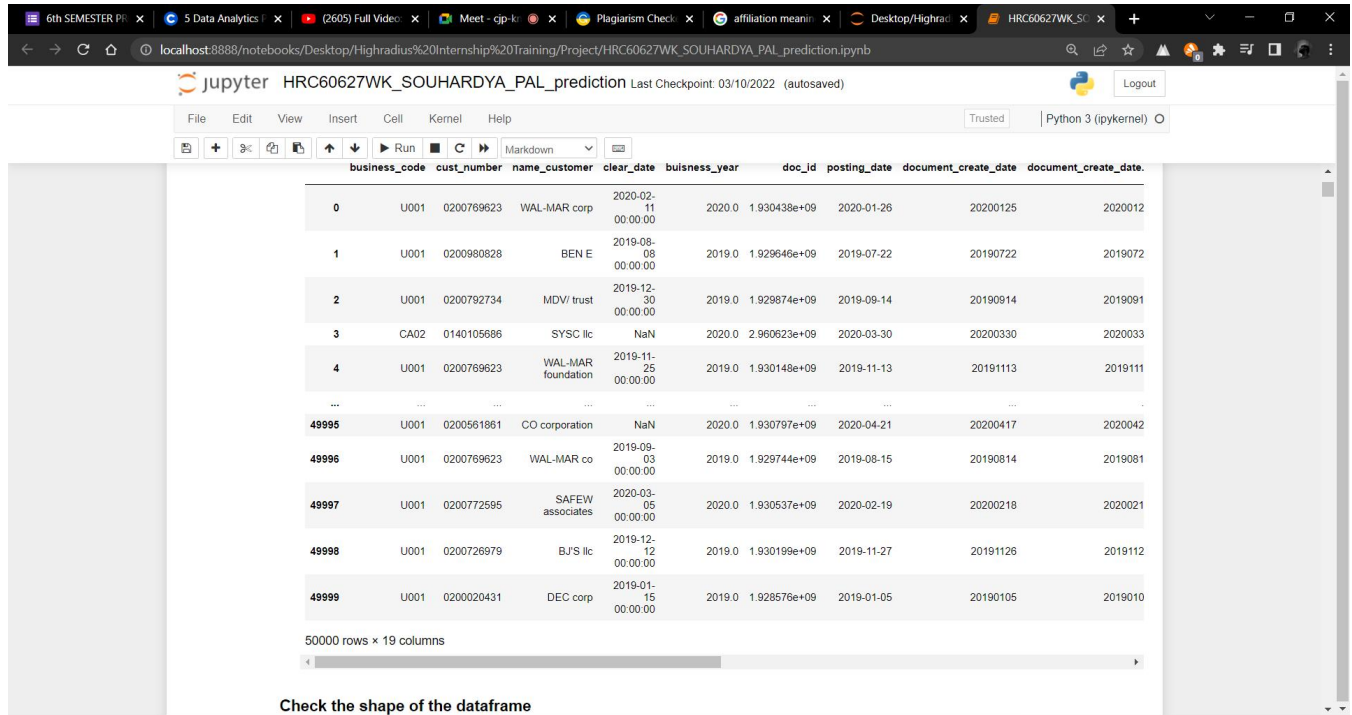
4.1.2 Collection Management

The creation of a prioritised worklist is the initial stage in the collection process. This worklist sets the sequence in which the collector will work on the open or past due bills or begin the dunning process (Dunning is the practise of methodically contacting with customers to ensure the recovery of accounts receivable). A/R teams have traditionally prioritised accounts based on passive characteristics such as ageing and invoice value. However, over the years, these lagging elements of delinquency have not produced sufficient results.

4.2 Discussion Of Project:

Considering the findings, the Invoice Management Application appears to be a more efficient method of managing invoices than manually reviewing each customer's invoice. The system considers thousands of invoices and then does regression analysis on them. The data reveal an RMSE of 2.4, which is quite low. It denotes that the Invoice Management Application is extremely efficient, with a 95% accuracy rate. This system can be used to track proper sales, estimate payment dates, delay, and prepone, among other things. The model's accuracy is 87.56 percent.

Some Screenshots of ML

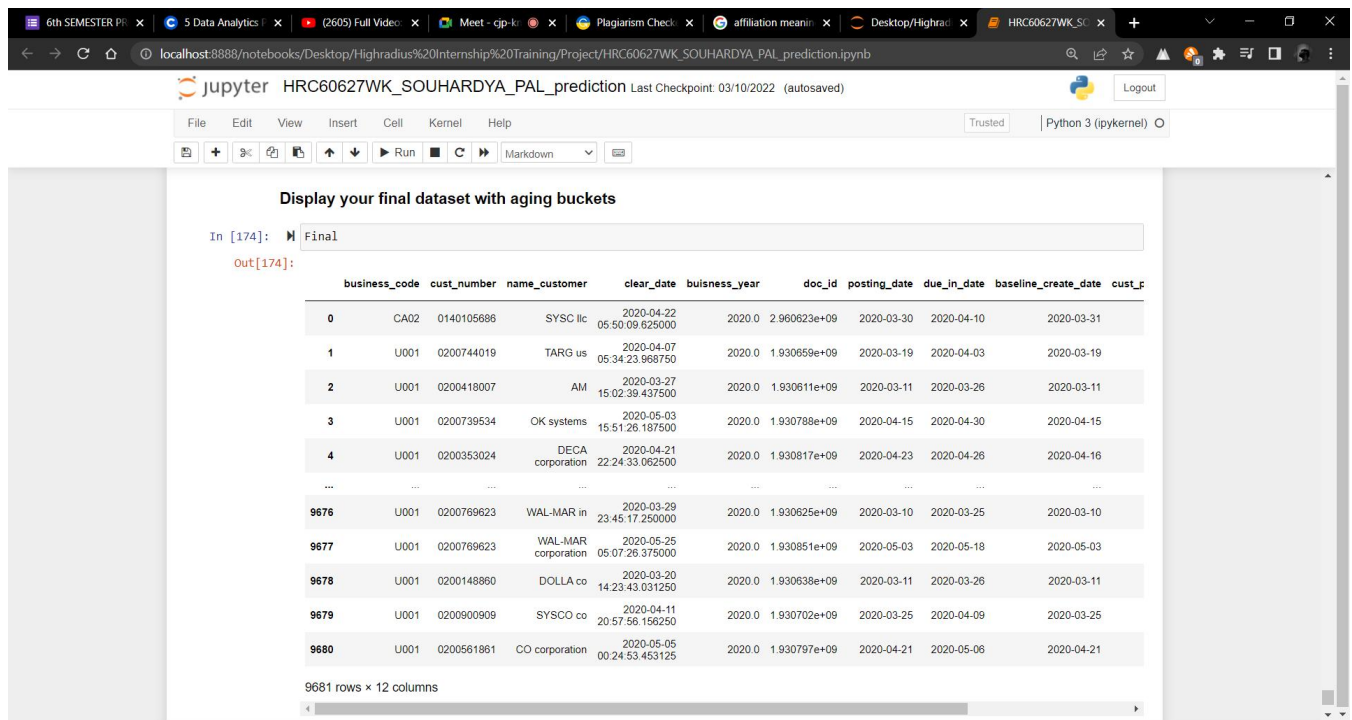


business_code cust_number name_customer clear_date buisness_year doc_id posting_date document_create_date document_create_date

0	U001	0200769623	WAL-MAR corp	2020-02-11 00:00:00	2020.0	1.930438e+09	2020-01-26	20200125	2020012
1	U001	0200980828	BEN E	2019-08-08 00:00:00	2019.0	1.929646e+09	2019-07-22	20190722	2019072
2	U001	0200792734	MDV/ trust	2019-12-30 00:00:00	2019.0	1.929874e+09	2019-09-14	20190914	2019091
3	CA02	0140105686	SYSC llc	NaN	2020.0	2.960523e+09	2020-03-30	20200330	2020033
4	U001	0200769623	WAL-MAR foundation	2019-11-25 00:00:00	2019.0	1.930148e+09	2019-11-13	20191113	2019111
...
49995	U001	0200561861	CO corporation	NaN	2020.0	1.930797e+09	2020-04-21	20200417	2020042
49996	U001	0200769623	WAL-MAR co	2019-08-03 00:00:00	2019.0	1.929744e+09	2019-08-15	20190814	2019081
49997	U001	0200772595	SAFEW associates	2020-03-05 00:00:00	2020.0	1.930537e+09	2020-02-19	20200218	2020021
49998	U001	0200726979	BJS llc	2019-12-12 00:00:00	2019.0	1.930199e+09	2019-11-27	20191126	2019112
49999	U001	0200020431	DEC corp	2019-01-15 00:00:00	2019.0	1.928576e+09	2019-01-05	20190105	2019010

50000 rows x 19 columns

Check the shape of the dataframe



business_code cust_number name_customer clear_date buisness_year doc_id posting_date due_in_date baseline_create_date cust_f

0	CA02	0140105686	SYSC llc	2020-04-22 05:50:09.625000	2020.0	2.960623e+09	2020-03-30	2020-04-10	2020-03-31
1	U001	0200744019	TARG us	2020-04-07 05:34:23.968750	2020.0	1.930659e+09	2020-03-19	2020-04-03	2020-03-19
2	U001	0200418007	AM	2020-03-27 15:02:39.437500	2020.0	1.930611e+09	2020-03-11	2020-03-26	2020-03-11
3	U001	0200739534	OK systems	2020-05-03 15:51:26.187500	2020.0	1.930788e+09	2020-04-15	2020-04-30	2020-04-15
4	U001	0200353024	DECA corporation	2020-04-21 22:24:33.062500	2020.0	1.930817e+09	2020-04-23	2020-04-26	2020-04-16
...
9676	U001	0200769623	WAL-MAR in	2020-03-29 23:45:17.250000	2020.0	1.930625e+09	2020-03-10	2020-03-25	2020-03-10
9677	U001	0200769623	WAL-MAR corporation	2020-05-25 05:07:26.375000	2020.0	1.930851e+09	2020-05-03	2020-05-18	2020-05-03
9678	U001	0200148860	DOLLA co	2020-03-20 14:23:43.031250	2020.0	1.930638e+09	2020-03-11	2020-03-26	2020-03-11
9679	U001	0200909099	SYSCO co	2020-04-11 20:57:56.156250	2020.0	1.930702e+09	2020-03-25	2020-04-09	2020-03-25
9680	U001	0200561861	CO corporation	2020-05-05 00:24:53.453125	2020.0	1.930797e+09	2020-04-21	2020-05-06	2020-04-21

9681 rows x 12 columns

Display your final dataset with aging buckets

In [174]: Final

Out[174]:

Screenshots of SQL

SQLyog Community 64 - [local/grey_goose - root@localhost]

File Edit Favorites Database Table Others Tools PowerTools Transactions Window Help

Filter tables in grey_goose

root@localhost

Tables

business

customer

writer_intership

Columns

Indexes

Views

Stored Procs

Functions

Triggers

Events

information_schema

my_db

mysql

performance_schema

sys

No nag screens on startup and shutdown : Reason #1 to upgrade

Query 1 Query 2 H2H_Intership* x History +

Messages Table Data Info

Limit rows First row 0 # of rows 1000

sl_no	business_code	cust_number	clear_date	business_year	doc_id	posting_date	document_create_date	document_create_date1	due_in_date	invoice_currency
1	U001	200769623	2020-02-11	2020	1930438491	2020-01-26	2020-01-25	2020-01-26	2020-02-10	USD
2	U001	200980828	2019-08-08	2019	1929646410	2019-07-22	2019-07-22	2019-07-22	2019-08-11	USD
3	U001	200792734	2019-12-30	2019	1929873765	2019-09-14	2019-09-14	2019-09-14	2019-09-29	USD
4	CA02	140105686	0000-00-00	2020	2960623488	2020-03-30	2020-03-30	2020-03-30	2020-04-10	CAD
5	U001	200769623	2019-11-25	2019	1930147974	2019-11-13	2019-11-13	2019-11-13	2019-11-28	USD
6	CA02	140106181	2019-12-04	2019	2960581231	2019-09-20	2019-09-20	2019-09-20	2019-10-04	CAD
7	U001	200769623	2019-11-12	2019	1930083373	2019-11-01	2019-10-31	2019-11-01	2019-11-16	USD
8	U001	200744019	0000-00-00	2020	1930659387	2020-03-19	2020-03-18	2020-03-19	2020-04-03	USD
9	U001	200769623	2019-06-18	2019	1929439637	2019-06-07	2019-06-05	2019-06-07	2019-06-22	USD
10	U001	200762301	2019-03-06	2019	1928819386	2019-02-20	2019-02-19	2019-02-20	2019-03-07	USD
11	U001	200418007	0000-00-00	2020	1930610806	2020-03-11	2020-03-06	2020-03-11	2020-03-26	USD
12	U001	200743129	2019-01-22	2019	1928550622	2019-01-02	2019-01-02	2019-01-02	2019-01-17	USD
13	U001	200186937	2019-05-06	2019	1929151655	2019-04-15	2019-04-15	2019-04-15	2019-04-30	USD
14	U001	200721222	2019-11-01	2019	1930022117	2019-10-17	2019-10-17	2019-10-17	2019-11-01	USD
15	U001	200739534	0000-00-00	2020	1930788296	2020-04-15	2020-04-15	2020-04-15	2020-04-30	USD
16	U001	200353024	0000-00-00	2020	1930817482	2020-04-23	2020-04-23	2020-04-23	2020-04-26	USD
17	U001	200794332	2019-11-12	2019	1930052739	2019-10-25	2019-10-25	2019-10-25	2019-11-09	USD
18	U001	200881076	2019-12-17	2019	1930209407	2019-12-02	2019-12-02	2019-12-02	2019-12-17	USD
19	U001	200769623	2019-11-26	2019	1930153511	2019-11-15	2019-11-14	2019-11-15	2019-11-30	USD
20	U001	200769623	2020-02-05	2020	1930438462	2020-01-24	2020-01-24	2020-01-24	2020-02-08	USD
21	U013	100053554	2020-02-11	2020	1991837617	2020-01-11	2020-01-07	2020-01-11	2020-02-10	USD
22	U001	200783734	2019-09-04	2019	1929773400	2019-08-21	2019-08-21	2019-08-21	2019-09-05	USD
23	U001	200744019	0000-00-00	2020	1930676042	2020-03-21	2020-03-20	2020-03-21	2020-04-05	USD
24	U001	100006311	2019-07-30	2019	1929626925	2019-07-17	2019-07-17	2019-07-17	2019-08-01	USD
25	U001	200769623	2020-02-04	2020	1930431304	2020-01-24	2020-01-23	2020-01-24	2020-02-08	USD
26	CA02	140106408	0000-00-00	2020	2960618790	2020-03-06	2020-03-06	2020-03-06	2020-03-16	CAD
27	U001	200769623	2019-01-30	2019	1928620435	2019-01-14	2019-01-14	2019-01-14	2019-01-29	USD
28	U001	200337148	0000-00-00	2020	1930592246	2020-03-02	2020-03-02	2020-03-02	2020-03-17	USD
29	U001	200769623	2019-05-09	2019	1929194820	2019-04-23	2019-04-23	2019-04-23	2019-05-08	USD

Database: grey_goose Table: writer_intership

Ready 1000 row(s) Ln 1, Col 39 Connections: 1 Upgrade to SQLyog Ultimate

Screenshots of UI

WebApp Final Project Submission x React App x +

localhost:3000

ABC Products highradius

Invoice List

PREDICT ANALYTICS VIEW ADVANCE SEARCH

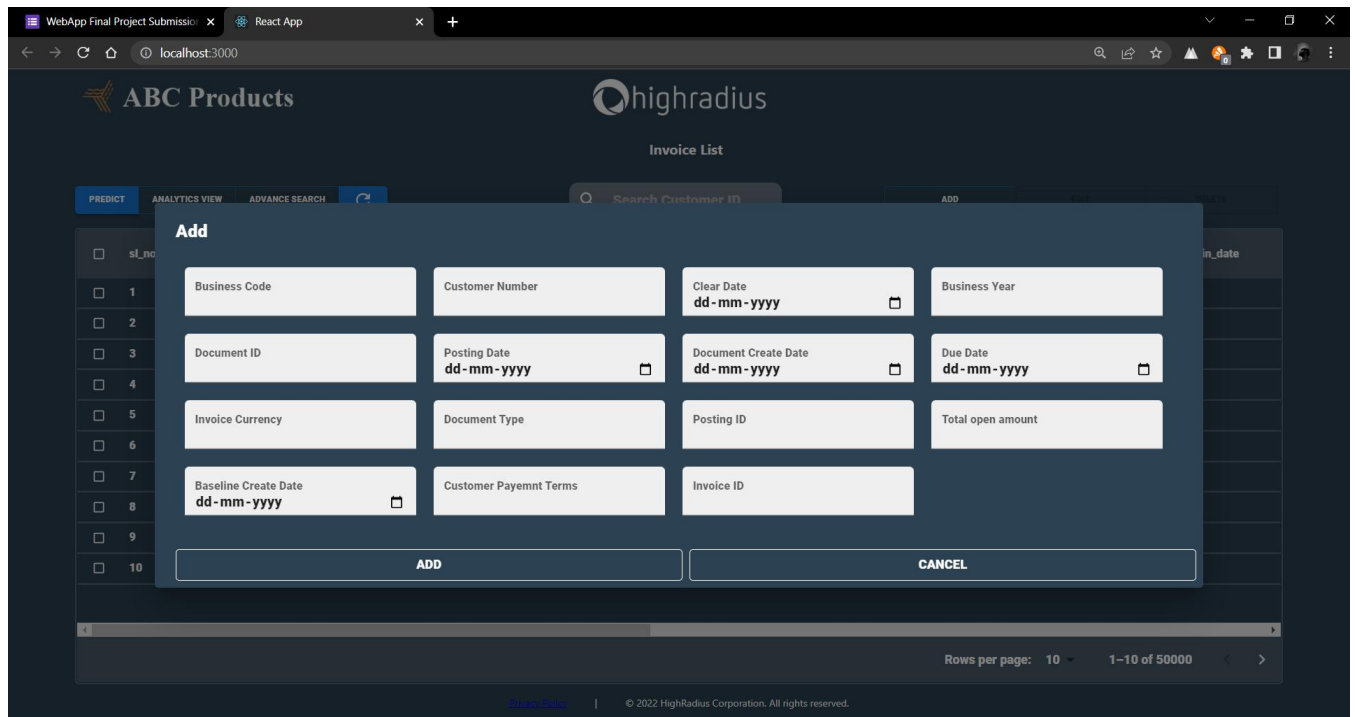
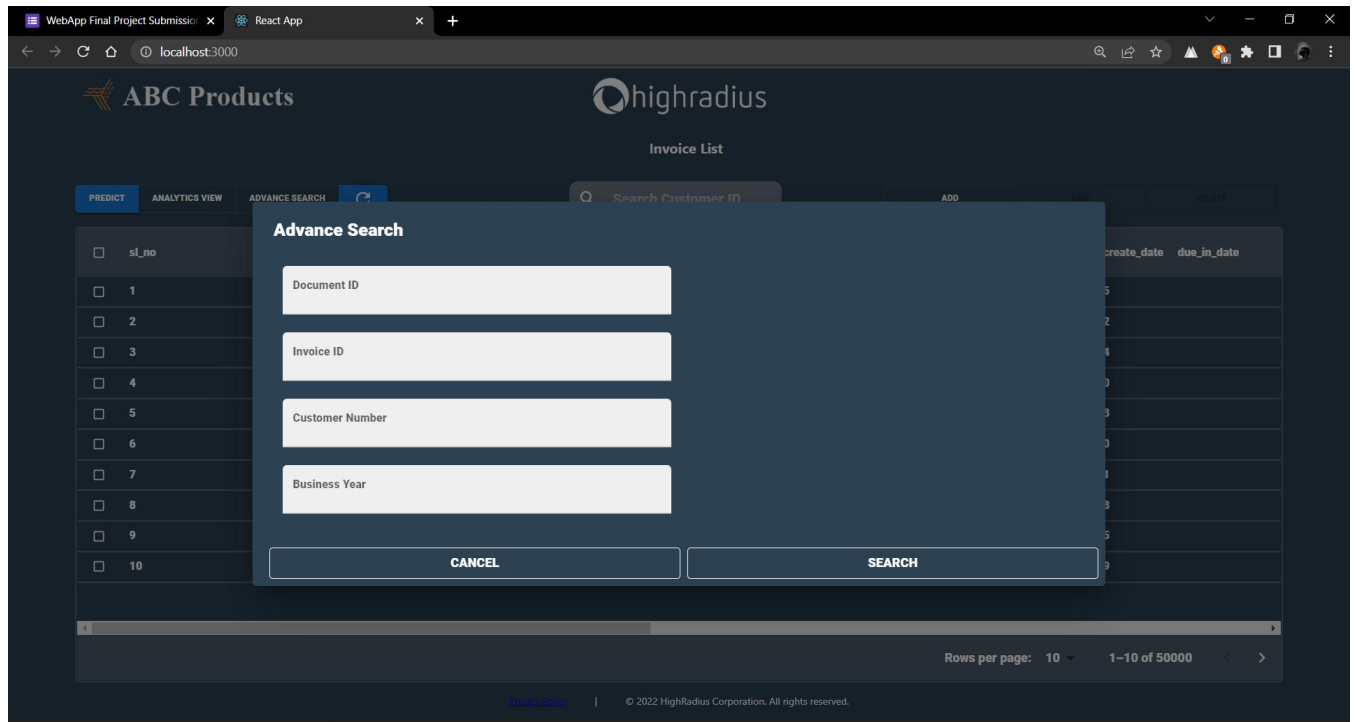
Search Customer ID

ADD EDIT DELETE

<input type="checkbox"/>	sl_no	business_code	cust_number	clear_date	business_year	doc_id	posting_date	document_create_date	due_in_date
<input type="checkbox"/>	1	U001	200769623	2020-02-11	2020	1930438491	2020-01-26	2020-01-25	
<input type="checkbox"/>	2	U001	200980828	2019-08-08	2019	1929646410	2019-07-22	2019-07-22	
<input type="checkbox"/>	3	U001	200792734	2019-12-30	2019	1929873765	2019-09-14	2019-09-14	
<input type="checkbox"/>	4	CA02	140105686	NA	2020	2960623488	2020-03-30	2020-03-30	
<input type="checkbox"/>	5	U001	200769623	2019-11-25	2019	1930147974	2019-11-13	2019-11-13	
<input type="checkbox"/>	6	CA02	140106181	2019-12-04	2019	2960581231	2019-09-20	2019-09-20	
<input type="checkbox"/>	7	U001	200769623	2019-11-12	2019	1930083373	2019-11-01	2019-10-31	
<input type="checkbox"/>	8	U001	200744019	NA	2020	1930659387	2020-03-19	2020-03-18	
<input type="checkbox"/>	9	U001	200769623	2019-06-18	2019	1929439637	2019-06-07	2019-06-05	
<input type="checkbox"/>	10	U001	200762301	2019-03-06	2019	1928819386	2019-02-20	2019-02-19	

Rows per page: 10 1-10 of 50000

© 2022 HighRadius Corporation. All rights reserved.



AI Enabled Fintech B2B Invoice Management Application

The screenshot displays the 'ABC Products' HighRadius 'Invoice List' interface. The application is running on a web browser at localhost:3000. The interface includes a header with the 'ABC Products' logo and 'highradius' branding. Below the header, there are tabs for 'PREDICT', 'ANALYTICS VIEW', and 'ADVANCE SEARCH', along with a search bar labeled 'Search Customer ID'. A table lists 10 invoices with columns: sl_no, business_code, cust_number, clear_date, business_year, doc_id, posting_date, document_create_date, and due_in_date. An 'Edit' modal is open over the first row (sl_no 1), containing fields for 'Invoice Currency' and 'Customer Payment Terms', and buttons for 'EDIT' and 'CANCEL'. The footer shows '1 row selected' and 'Rows per page: 10'.

sl_no	business_code	cust_number	clear_date	business_year	doc_id	posting_date	document_create_date	due_in_date
1	U001	200769623				2020-01-26	2020-01-25	
2	U001	200980828				2019-07-22	2019-07-22	
3	U001	200792734				2019-09-14	2019-09-14	
4	CA02	140105686				2020-03-30	2020-03-30	
5	U001	200769623				2019-11-13	2019-11-13	
6	CA02	140106181				2019-09-20	2019-09-20	
7	U001	200769623				2019-11-01	2019-10-31	
8	U001	200744019				2020-03-19	2020-03-18	
9	U001	200769623	2019-06-18	2019	1929439637	2019-06-07	2019-06-05	
10	U001	200762301	2019-03-06	2019	1928819386	2019-02-20	2019-02-19	

This screenshot is identical to the one above, showing the 'ABC Products' HighRadius 'Invoice List' interface with the 'Edit' modal open for the first row. The interface includes the same header, tabs, search bar, table, modal, and footer.

sl_no	business_code	cust_number	clear_date	business_year	doc_id	posting_date	document_create_date	due_in_date
1	U001	200769623				2020-01-26	2020-01-25	
2	U001	200980828				2019-07-22	2019-07-22	
3	U001	200792734				2019-09-14	2019-09-14	
4	CA02	140105686				2020-03-30	2020-03-30	
5	U001	200769623				2019-11-13	2019-11-13	
6	CA02	140106181				2019-09-20	2019-09-20	
7	U001	200769623				2019-11-01	2019-10-31	
8	U001	200744019				2020-03-19	2020-03-18	
9	U001	200769623	2019-06-18	2019	1929439637	2019-06-07	2019-06-05	
10	U001	200762301	2019-03-06	2019	1928819386	2019-02-20	2019-02-19	

The screenshot displays a web application interface for 'ABC Products' and 'highradius'. The main section is titled 'Invoice List'. It features a table with columns: sl_no, business_code, cust_number, clear_date, buisness_year, doc_id, posting_date, document_create_date, and due_in_date. A modal dialog titled 'Delete Records?' is open, asking 'Are you sure you want to delete these record[s]?' with 'CANCEL' and 'DELETE' buttons. The table shows 10 rows, with 4 rows selected. The footer includes '© 2022 HighRadius Corporation. All rights reserved.'

sl_no	business_code	cust_number	clear_date	buisness_year	doc_id	posting_date	document_create_date	due_in_date
1	U001	200769623	2020-02-11	2020	1930438491	2020-01-26	2020-01-25	
2	U001	200980828	2019-08-08	2019	1929646410	2019-07-22	2019-07-22	
3	U001	200792734				2019-09-14	2019-09-14	
4	CA02	140105686				2020-03-30	2020-03-30	
5	U001	200769623				2019-11-13	2019-11-13	
6	CA02	140106181				2019-09-20	2019-09-20	
7	U001	200769623	2019-11-12	2019	1930083373	2019-11-01	2019-10-31	
8	U001	200744019	NA	2020	1930659387	2020-03-19	2020-03-18	
9	U001	200769623	2019-06-18	2019	1929439637	2019-06-07	2019-06-05	
10	U001	200762301	2019-03-06	2019	1928819386	2019-02-20	2019-02-19	

Chapter 5

5.1 Conclusion:

Experiences from various industries show that companies may have problems collecting customer invoice payments. Our aim was to understand customer behavior regarding invoice payments, and propose an analytical approach to learning and predicting payment behavior. Our logic can then be embedded into a system where decision makers can make predictions regarding future payments, and take actions as necessary toward the collection of potentially unpaid debt, or adjusting their financial plans based on the expected invoice-to-cash amount.

It is critical to identify the system requirements by appropriately gathering required data to interact with the system in order for the software to perform successfully. We gained a lot of practical information from this project, which we believe will help us to be in a better position in the future.

5.2 FUTURE WORK:

Organizations, firms, and even individuals may keep track of the payment of their orders with the aid of AI Enabled Fintech B2B INVOICE MANAGEMENT Application, regardless of the big number of clients. This will not only save time, but will also provide a functional environment for an organization's and/or workplace's proper functioning.

References

- [1] <https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/>
- [2] <https://reactjs.org/>
- [3] <https://www.javatpoint.com/servlet-tutorial>
- [4] [How to Choose a Feature Selection Method For Machine Learning \(machinelearningmastery.com\)](#)
- [5] [Java Database Connectivity\(JDBC\) \(knowledgehut.com\)](#)

TURNITIN PLAGIARISM REPORT

(This report is mandatory for all the projects and plagiarism must be below 25%)

Sample_turnitin_report_for_students.docx			
ORIGINALITY REPORT			
13%	7%	3%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	Submitted to Kennesaw State University		3%
	Student Paper		
2	www.guardian.co.uk		2%
	Internet Source		
3	Campbell, Neil. "Post-Western Cinema", A Companion to the Literature and Culture of the American West Witschi/A Companion to the Literature and Culture of the American West, 2011.		1%
	Publication		