SMU Data Science Review

Volume 2 | Number 3 Article 7

2019

Identifying At-Risk Clients for XYZ Packaging, Co.

Eduardo Carlos Cantu Medellin
Southern Methodist University, ecantumedellin@mail.smu.edu

Mihir Parikh
Southern Methodist University, mparikh@smu.edu

Christopher Graves
ccgraves@smu.edu

Brendon Jones
XYZ Packaging, bjones@xyzpackaging.com

Identifying At-Risk Clients for XYZ Packaging, Co.

Carter Graves[1], Mihir Parikh[1], Eduardo Cantu Medellin[1], and Brendon Jones[2]

[1] Master of Science in Data Science, Southern Methodist University, Dallas TX 75275 USA {ccgraves,mparikh,ecantumedellin}@smu.edu
[2] XYZ Packaging Inc., 8350 Star Way, Oklahoma City, OK 73008 USA bjones@xyzpackaging.com

Abstract. We present a multi-algorithmic modeling approach for the identification of at-risk customers for XYZ Packaging Inc. We define at-risk customers as those having declining seasonally adjusted gross income forecasts which are a strong indicator of impending customer churn. Customer retention is an area of interest regardless of industry but is especially vital in commodity-based low margin industries. We employ traditional Autoregressive Integrated Moving Average (ARIMA) and Anomaly Detection algorithms for discriminating changes in customer revenue patterns. Ultimately, we identify a meaningful proportion of clients whose forward-looking quarterly demand can be predicted within an actionable degree of accuracy.

1 Introduction

XYZ Packaging, Inc. (XYZ) provides custom packaging solutions for a diverse client base. Clients range in size from small to medium size businesses up to and including Fortune 500 companies. These clients also vary widely across industries, including Energy, Heavy Industrial, Medical, and government entities. The packaging industry is a highly saturated market. Thus, competitors in this space operate on thin profit margins. Due to the custom nature of their product offerings, XYZ engages in a significant pre-sale activity with prospective clients. This translates to high customer acquisition costs relative to competitors providing standardized products. This combination of high customer acquisition costs and low-profit margins means XYZ is intensely focused on customer retention, the reduction of churn, in order to recoup acquisition costs and realize profits. XYZ has been an operating company for thirty years. During the preceding fifteen years, they have input and fulfilled all orders through an in-house Enterprise Resource Planning (ERP) system.

XYZ is unable to reliably forecast customer demand and identify clients at-risk of loss. The prevailing view within XYZ executive management is that their customers predominantly operate as "job shops". That is, demand for XYZ's products is driven by limited or one-off projects. According to this view, historical order data is not useful for forecasting future demand on a per client basis. Thus, it cannot be relied upon to identify at-risk clients. Operating from

this vantage, customer demand forecasting and identification of at-risk clients is generated solely from conversational intelligence gathered by sales representatives. XYZ sought to test a countervailing view of their client's demand. That is, a meaningful percentage of clients do exhibit predictable patterns of demand. However, these patterns are occluded by the proportion of clients that present pseudo-random behavior. Further, they are predicted by relations within the historical behavior pattern more complex than simple measures of autoregressive revenue timing. Advanced modeling techniques can be used to make predictions about a portion of their client portfolio within an actionable level of confidence.

In order to provide XYZ with a method for identifying predictably at-risk clients within this population, we were provided with ten years of detailed client ordering behavior, also we define at-risk as the expectation of a decline in seasonally adjusted gross income in the following calendar year. The obtained data is granular to the item, customer, and invoice level. It includes attributes relative to timing, price, product cost, freight costs, product mix, order fulfillment locations, raw material consumption, and order fulfillment labor. It is important to note that all items within the dataset are customer specific. We wished that our models draw upon any predictive ability at the item level that exists across customers. For this reason, we sought to derive features relative to the change in product mix over time and the underlying raw materials of which these products were comprised. In consultation with XYZ management, we identified and derived additional features from the base data. These fell into the categories of input costs, gross income, sales order velocity, fulfillment times, product mix, product change velocity.

Guided by XYZ management expertise and experience, we knew that stand alone autoregressive (AR) and moving average (MA) modeling approaches built upon revenue by client and period had been unsuccessful in discriminating pattern following clients. Assuming non-stationarity in the data we began our efforts by building an autoregressive integrated moving average (ARIMA) model fitted upon a combination of the base and derived data points discussed above. This model was used to make quarterly (three months) revenue predictions for ten randomly selected quarters within each client life cycle. The subset of customers with high cross fold validation accuracy scores were then compared against a two-dimensional principal component analysis of the feature space. These approaches confirmed the existence of two distinct clusters within the client deck. Thirty-three percent of clients presented average cross fold revenue forecast accuracy with non statistically significant difference from pure guessing. However, the remainder hewed to within 80% of the actual average gross income.

We selected 80% of the gross income as our threshold to identify the top customers and focus on their analysis. We identified 47 customers within this threshold, which we then applied the ARIMA models and forecast their revenue trend without the seasonality for the next quarter. Based on the information provided by the ARIMA model then we evaluated if the trend was declining or stable in order to flag the customer at-risk. After the ARIMA procedure, we used the Ruptures library that provided us the ability to identify the step function

variation during the time series which could have influenced the outcome of the ARIMA model.

Our process was able to identify at-risk customers based on their seasonally adjusted gross income trend and yielded a model with 77% accuracy. With this information we can provide the company an early warning detection for customer churn within the subset of customers that comprise 80% of their gross income. The remainder of this paper is organized as follows. In the next section, two, we provide an overview of the modeling approach and experiments performed along with the rationale for their use. Section three provides a detailed description of the data set, including the base and derived features along with pre-modeling exploratory analysis. Also, it shows specifics of the modeling steps and techniques employed. Section four is an overview of a Cloud deployment architecture that is readily scaleable and highly available. Section five goes over the results of our predictions. Section six highlights some ethical considerations of our work. Lastly section seven provides an overview of our conclusions.

2 Forecasting Procedures

Time series forecasting models are based on the fundamental concept that outputs are impacted by previous outputs. This means that each output is not independent and is correlated to an output value at a previous time point. There are several industry standard modeling techniques used for forecasting time series data such as autoregressive integrated moving average (ARIMA) models.

2.1 ARIMA

ARIMA models are often used as a first pass because they are a relatively simple linear modeling techniques whose features can often be understood. ARIMA techniques have been extended to accommodate some non-linear patterns, however, are limited in overall application due to their specificity. For many applications, an ARIMA model may provide enough practical predictive accuracy that addressing non linear components may not be worth the extra investment.

An ARIMA model is composed of auto regression, moving average, and a difference factor. The predicted value in an ARIMA function is based off of a linear relationship between past observations and random error. ARIMA models are characterized by the tuning of three key parameters d, p, and q. The variable d represents the differencing required to make time series data stationary. Stationary refers to a constant mean and variance thus removing the dependence on the time observation itself by eliminating trend and seasonality. The variable p represents the number of lagged previous periods observations considered in the autoregressive calculation. Finally, the variable q represents the number of lagged previous forecast periods where errors cannot be explained by trend [4].

2.2 Change Point Detection

XYZ wishes to understand not just which clients are at risk, but also where in time they may have begun an unfavorable trend. While our data is time series, there are many different trends to evaluate. The change point detection algorithm *ruptures* in python was designed for non-stationary signals, of which time series is a good example. The methods included in the package cover exact and approximate detection for various parametric and non-parametric models.

We focused on the window-based method provided in the *ruptures* package. This method is intended to perform an approximation on the change point, making the signal segmentation efficient and fast [3]. The algorithm uses two sliding windows across the data.and applies a cost function for each of the windows and then calculates the discrepancy in order to compare the two windows as follows:

$$d(y_{u..v}, y_{v..w}) = c(y_{u..w}) - c(y_{u..v}) - c(y_{v..w})^1 \tag{1}$$

Where $\{y_t\}_t$ is the input signal and $u < v < w$ are the indexes from the data. The cost is calculated throughout the data in a sliding window. If the window segments $u..v$ and $v..w$ fall into similar cost sections then the discrepancy between the first and second window is low. However, if these two segments fall into different cost sections the discrepancy between the windows will be significantly higher [2]. Figure 1 shows how cost can differ between sliding windows.

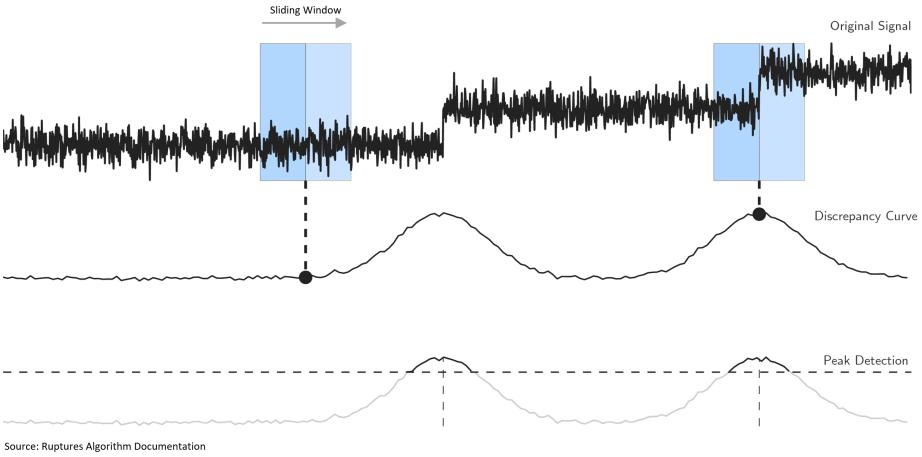

Fig. 1. Differences in cost for a sliding window. The chart shows a window split into two sections, a darker and a lighter blue section. The window slides across the signal comparing each section and looking for the delta to cross the detection threshold. Once the threshold is reached the algorithm will return the index where the step function happened on the time series.

[1] http://ctruong.perso.math.cnrs.fr/ruptures-docs/build/html/detection/window.html

The output from the algorithm provides the data indexes where the change happens. This information is useful to our problem because it allows us to locate large differences between the two segments of the time series. The time locations provided are the insight to pin-point when the problem started with the client.

3 Data Pipeline

The idea of the general model is to have a process to identify top customers that make up 80% of the total revenue. Once this was complete, we created a forecast for the next three months and analyzed the trends. Customers with declining seasonally adjusted gross revenue were classified as at-risk and marked as an opportunity to take action. Customers with upward trending gross revenues were identified for further analysis for sales and account managers to understand best practices to potentially apply to other customers. Figure 2 illustrates the key elements of our pipeline.

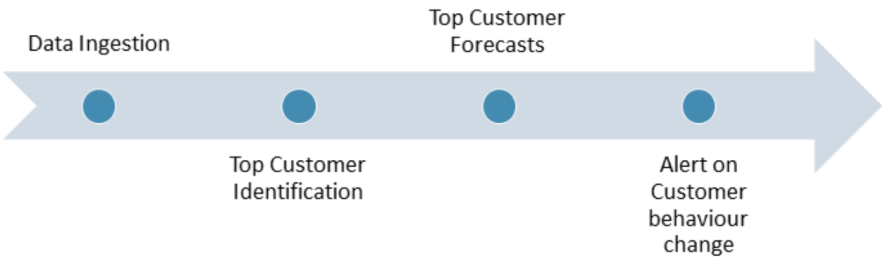

Fig. 2. Data pipeline for customer alerts. High level process with the stages to identify an at-risk client

3.1 Data Ingestion

XYZ provided a dataset with 10 years of itemized orders which was the result of joining three tables:

- Receivables: This table has information on the order level and its cost of handling it.
- ReceivablesItems: The information is on the item level for this table. Therefore, the relation between 'Receivables' and this table is of one-to-many.
- SalesOrder: This describes labor costs to produce the item and how it is released to the customer.

There are 24 features included in the dataset. The features of interest are the 'Ship Date', 'CustomerNo', 'Price each', 'Qty', and 'MaterialCost'. These

features are the preliminary focus area, and as the analysis progresses, other features can be integrated as well. There are two types of data present in the files, eighteen categorical fields and seven numerical fields. One calculated field 'GrossIncome' was added:

$$GrossIncome = PriceEach - MaterialCost \tag{2}$$

A total of 1,289 customers are included in the dataset as indicated by the 'Customer#' field. The top customer is 'A15216' with 80,489 items ordered during the collected period. Additional information present is the shipping location. There are six central locations where the company ships its products of which nearly half are sent to Austin. Table 1 shows some of the features included in the dataset.

Table 1. Features Included In Dataset

Field Name	**Table**	**Type**	**Description**
InvoiceNo	Receivables	String	Accounting Invoice Number
InvoiceType	Receivables	String	Invoice Type
CustomerNo	Receivables	String	Customer Number
CustomerName	Receivables	String	Customer Name
ShipAddress	Receivables	String	Ship To Address
Zip	Receivables	String	Ship To Zip Code
Fabricated_MC_Total	Receivables	Float	Fabricated Material Cost
FGHO_MC_Total	Receivables	Float	Finished Goods Material Cost
BillofLading	Receivables	String	Bill of Lading for Shipping
Freight	Receivables	Float	Freight / Shipping Cost
Location	Receivables	String	Location of WIC Branch Office
ShipDate	Receivables	Date	Ship Date
PN	Receivables	String	Internal Part Number
Description	ReceivableItems	Text	Part Description
WONumber	ReceivableItems	String	Work Order Number
Qty	ReceivableItems	Int	Quantity Shipped
QtyDescription	ReceivableItems	String	Quantity Shipped Description
SRV	ReceivableItems	Boolean	Labor Used Flag
FGHO_Item	ReceivableItems	Boolean	Finished Goods Handling Only Flag
PriceEach	ReceivableItems	Float	Price Each
MaterialCost	ReceivableItems	Float	Material Cost to produce Item
HeatTreatedLumber	SalesOrder	Boolean	Heat Treated Lumber Flag
Labor	SalesOrder	Float	Labor Cost to Produce Item
ReleaseType	SalesOrder	String	How to release to customer

3.2 Data Exploration

Duplication analysis found 1,300 duplicated rows. For any two or more rows where all columns were equal to the corresponding duplicate, only one copy of

the row was retained for this analysis. The number of records left after removing the duplicates was 538,567. Also, SRV and FGHO features have identical values of False for all records hence both columns were removed. There are two rows missing data in the fields used to calculate the gross income. The amount of missing data in the fields of interest is not enough to have an impact on the modeling and predictive work that needs to be completed. However, there are some categorical fields, like 'Release Type', that have more than half of their values missing. There were ten years worth of data; however, in the dataset, there were some records that showed dates before and after the valid dates. Therefore, we removed all the records before 01/01/2009 and after 12/31/2018 from the set. In addition to eliminating the times outside the given range, we also removed dates with the value of '00/0/00' from the dataset. As of now, the company has not provided any data for the year 2019. Any data for 2019, can be used to validate the predictive model.

In the data we found negative values for the 'Price each' variable, which was causing problems in the data aggregation (table 2). Such records were removed to avoid having wrong estimates on gross income. After eliminating these records, other variables that had negative values like 'Qty' did not present negative values anymore (table 3).

Table 2. Including negative values

	count	mean	std	min	25%	50%	75%	max
FGHO Mat Cost	534662.0	388.3	1943.5	-594000.0	0.0	0.0	311.4	594000.0
Freight	534661.0	1.0	18.5	-1716.3	0.0	0.0	0.0	2720.3
Qty	534660.0	304.8	5657.7	-267300.0	2.0	18.0	70.0	944460.0
Price each	534660.0	84.5	934.3	-373437.8	2.7	13.8	53.5	373437.8
MaterialCost	534660.0	33.0	207.8	-1650.0	1.1	5.6	21.5	41175.5
Labor	534660.0	1.3	7.8	0.0	0.0	0.0	0.0	1016.5
GrossIncome	534660.0	321.6	1785.9	-650160.0	44.3	134.4	338.4	650160.0

Table 3. Negative values removed

	count	mean	std	min	25%	50%	75%	max
FGHO Mat Cost	486494.0	433.8	1739.9	-303.3	0.0	0.0	380.6	594000.0
Freight	486494.0	1.1	18.6	0.0	0.0	0.0	0.0	2720.3
Qty	486494.0	340.6	5816.7	0.0	5.0	20.0	80.0	944460.0
Price each	486494.0	71.7	144.5	0.0	3.5	15.0	56.9	1350.0
MaterialCost	486494.0	29.5	61.9	0.0	1.6	6.7	25.5	2981.0
Labor	486494.0	1.4	7.2	0.0	0.0	0.0	0.0	1016.5
GrossIncome	486494.0	342.4	1181.7	-64530.0	59.0	151.2	361.7	650160.0

After the cleaning process of the dataset, the number of records remaining for analysis and modeling is 486,694.

3.3 Top Customer Identification

The process to identify the customers that contribute the most to the company is based on its contribution to the monthly aggregation of the 'GrossIncome' variable. To start, we aggregated the data monthly to identify the high level gross income variable overall trend.

Several variables had their trends explored: 'Price each', 'Qty', 'Total of Orders', and 'GrossIncome'. This section of the document only focuses on 'GrossIncome' trend for the company by customer level. For the latter, we only take a look at the top customers.

First, the gross income calculation was done monthly, in which all sales for all customers were aggregated by month then plotted to visualize over time. As shown in figure 3, monthly gross income has grown over time. Since 2009, after the 2008 recession, it started to climb at a steady pace until 2016, when it began to rise at a faster pace.

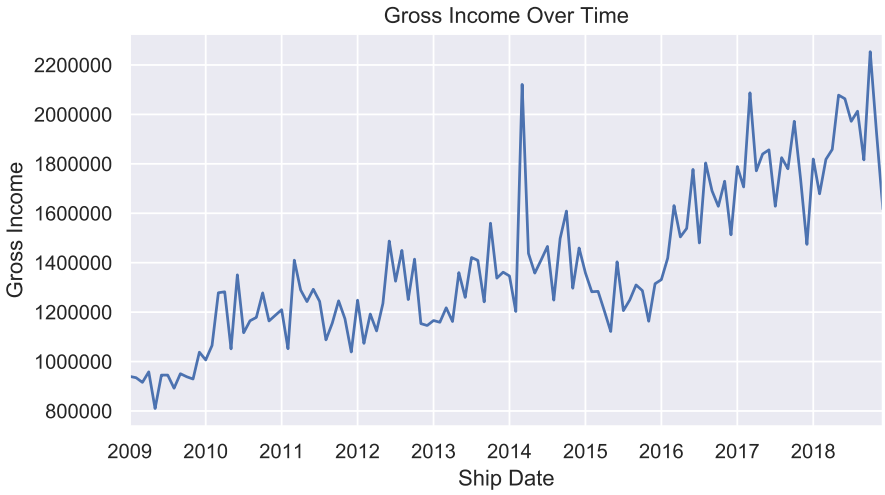

Fig. 3. Monthly Gross Income Aggregate for all customers.

To better understand the overall trend, a look into the seasonal and random components of the pattern makes clear where the trend is going (figure 4). There is seasonality of one year, having the last and first month of the year as the low point for gross income, and Q2 and Q3 the quarters with the most income every year. Also, there is a random component to the trend that largely has been

maintained within the same range across all the years. It appears that 2012 was an outlier with significant departure from prevailing year over year and seasonal trends.

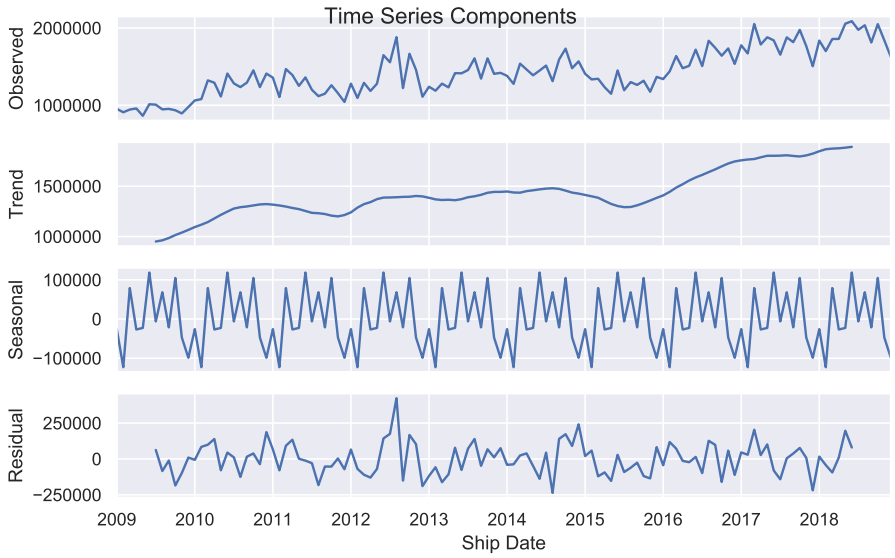

Fig. 4. Margin trend breakdown. First chart shows the original gross income time series. Second chart shows the trend component of the time series. Third chart plots the seasonality part of the time series. Last chart plots the random component of the time series.

Second, by looking at the previous trend we find that since 2015 there is an upward trend in the income. Considering this upward trend, we have decided to narrow the time window for our customer search from 2015 to the end of 2018. Within this period, we searched for the customers that contribute the most to the trend. By aggregating all the sales per customer during this search window, we created a cumulative distribution function (Figure 5) and found that the elbow started around 80% of the total gross income. This suggests that only a few customers contribute to the majority of the gross income. The other 20% of the gross income is distributed by low volume customers.

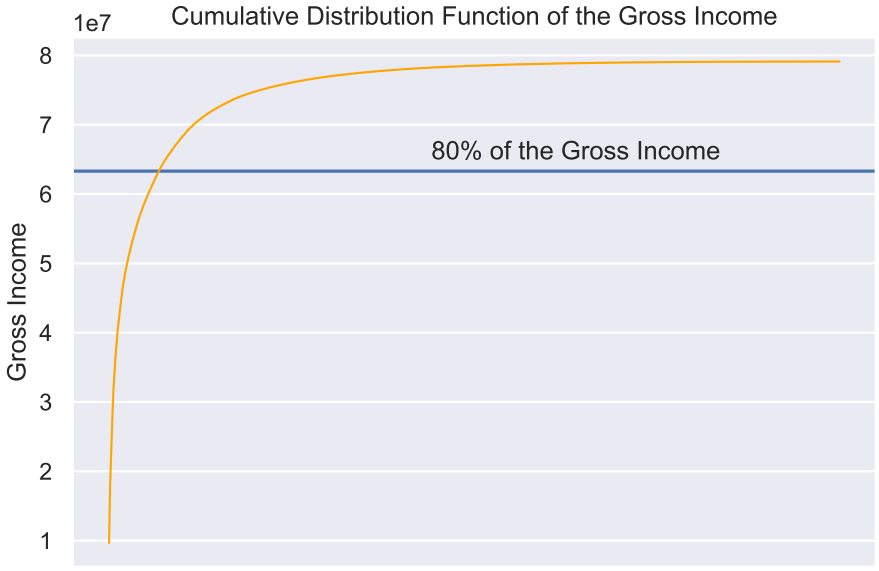

Fig. 5. Gross Income Cumulative Distribution Function for all Customers

Using the criteria from the cumulative distribution function, we found 47 customers contributing to 80% of the gross income. Figure 6 shows the top 47 customers found through this process. The top customer, '210' in itself contributes 12% of the gross income.

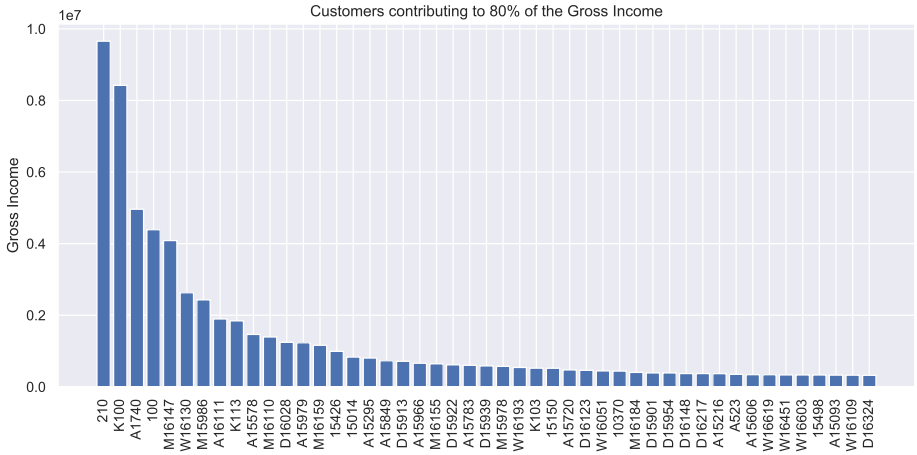

Fig. 6. Top 47 customers that contribute to 80% of the Gross Income.

3.4 ARIMA Modeling

We used the aggregated gross income for the top 47 customers and performed a grid search to obtain the best ARIMA model. The grid search included the parameters p, d, q and a seasonal differencing component. Each of the values for p, d, q could take the values of 0 and 1, as well for the seasonal differencing component. Although technically these parameters can take values greater than 1, it was decided to limit the grid search. This decision was based on consistent seasonal pattern observed on figure 4, in which we do not want to have more than one order of seasonal differencing or more than two orders of total differencing (seasonal + none-seasonal) [2]. The search model looked for the best ARIMA *(p,d,q) x (P,D,Q)*, where p is the number of autoregressive terms, d is the number of nonseasonal differences needed for stationary signal, and q is the number of lagged forecast errors in the prediction. And for the seasonal differencing component we have P, the number of seasonal autoregressive terms, D, the number of seasonal differences and Q the number of seasonal moving average terms.

We selected the ARIMA model with the lowest AIC provided by the grid search. The model with the lowest AIC was ARIMA (0, 1, 1) x (0, 1, 1). The AIC value was 2,508, and to provide context of this number, the highest AIC for the grid search was 3,602. The reason we selected the AIC (Akaike's Information Criteria) as a metric was that it can be use to compare a set of statistical models to each other. This metric is relative to within the set of models, and the lowest AIC value is ranked as the best model of the set.

Figure 7 does not show a departure from normality on the residuals. We can see that the theoretical quantiles are predominantly linear with the exception of a few outliers at both edges of the graph. We also observed that the lags do not go above 0.25 indicating that the differentiation term can make it stationary.

[2] https://people.duke.edu/ rnau/arimrule.htm

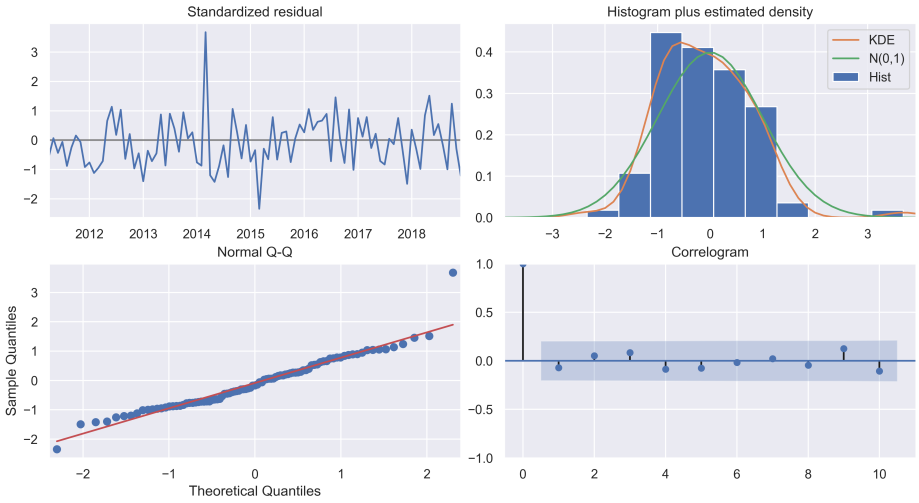

Fig. 7. Diagnostic plots for ARIMA.

Once the top customers are identified, a seasonally adjusted forecast is created for each of the top customers. Then based on the trend it was determined if there was any expected gross income decline by evaluating the trend direction.

3.5 Alert on Customer Behavior

The key behavior to recognize is a customer at risk. This is a customer who is expected to decline in seasonal adjusted gross income over the following calendar year. To obtain this we applied the ARIMA model obtained with the top customer aggregated data using the observations from 2009 to 2017 to forecast gross income for the first three months of 2018, and then compared the trend with the observations during the same period. If the trend indicates a drop in gross income for the next three months of 2018 the customer is flagged as at risk. Figure 8 shows one customer's complete time series observation depicted by the dark blue line along with a magnified section on the Forecast showing the direction for the seasonally adjusted and non-adjusted trends. The magenta line in the magnified section of the figure depicts the first three months of 2018 seasonal adjusted forecast in decline, which is in line with the observation series (dark blue line).The red line shows the un-adjusted trend moving upward, which can be misleading due to the seasonality. As per the observations there is a decline in gross income for this customer. Therefore, removing the seasonality from the forecast will provide a better estimate of customer behavior.

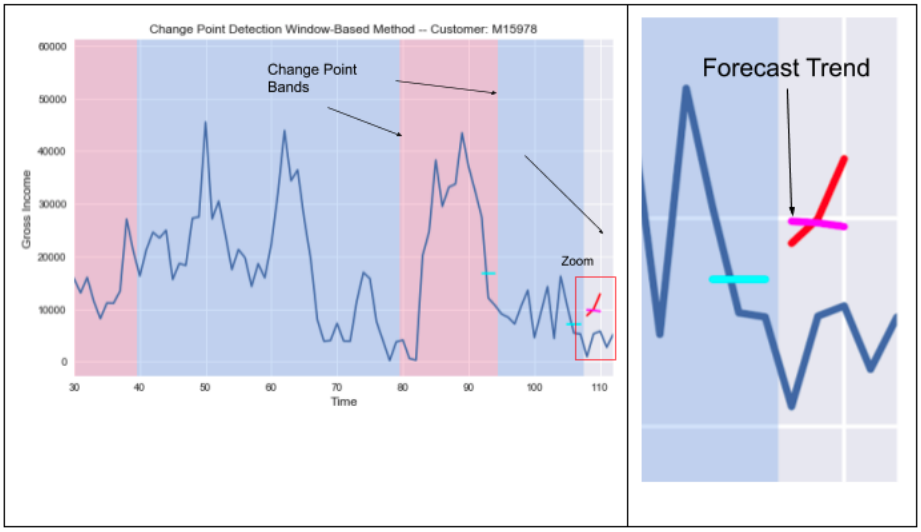

Fig. 8. At risk customer declining prediction trend.

The second behavior has to do with a forecast that is stable and does not need to be flagged. This customer can have an upward or flat trend in revenue (Figure 9). In contrast with figure 8, the seasonally adjusted trend for this customer is moving upwards as shown by the direction of the magenta line. This trend does not represent a concern for the company, however can provide insight in best practices that lead to these outcomes.

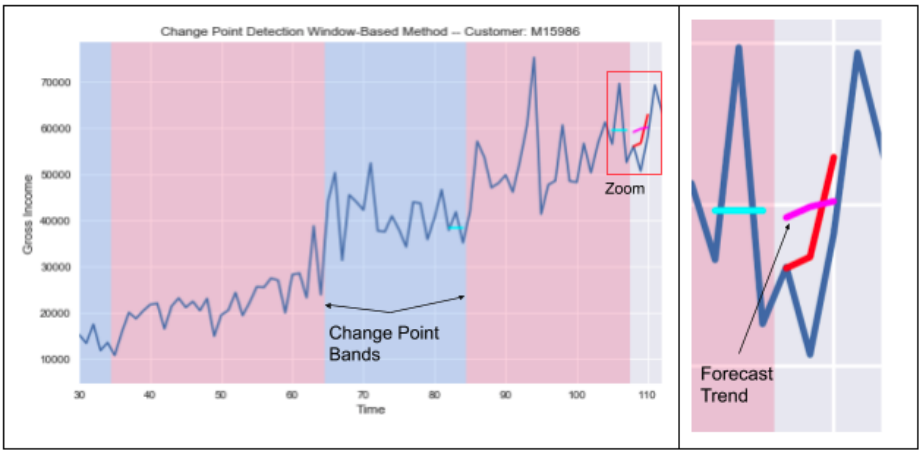

Fig. 9. A customer with a stable prediction trend.

3.6 Anomaly Detection

Once the forecast was complete for each of the customers the change detection algorithm was applied to evaluate large shifts in the time series. This information is useful to the sales team. They can get the month when the client started to deviate from the growth pattern. There are two outcomes as a result of this step, the first one is a step function in declining revenue, and the second is in the opposite direction with a step up in revenue. In both cases, this information is valuable for the sales team.

The algorithm used for determining the change point over the time series is a window-based change point detection method, which was described in the background section. We used the *ruptures* library which provides a method for identification.

We took one of the customers as an example and applied the algorithm to provide the changes in revenue throughout the period.

The shaded sections in Figure 10 indicate changes in the revenue in the time series. We can see that the first change happens around the observation #30 and then there is another change around observation #80. With this information we can calculate the difference in revenue for the last two periods of change and evaluate whether the change was positive or negative.

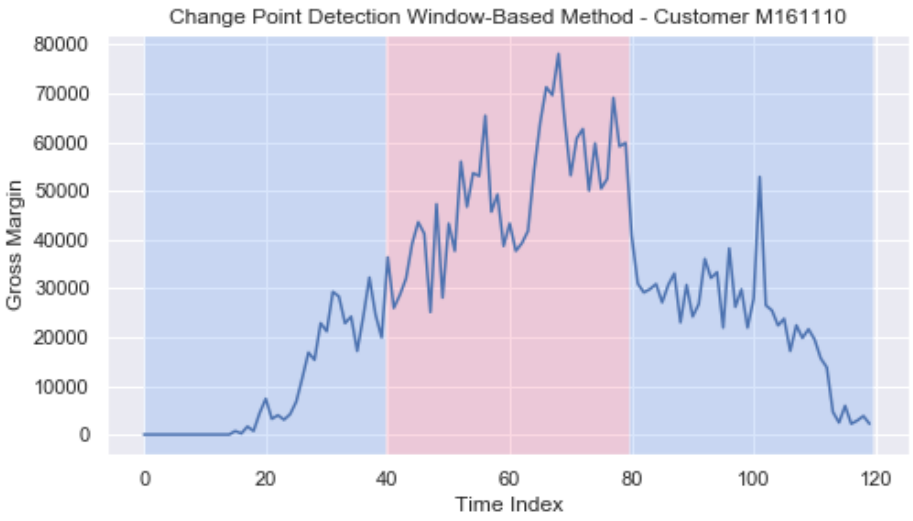

Fig. 10. Change point detection window

Figure 11 shows an increase in revenue, for this customer we see that there is a change from the initial trend, which seems to flatten for a few months from observation #65 to #85.

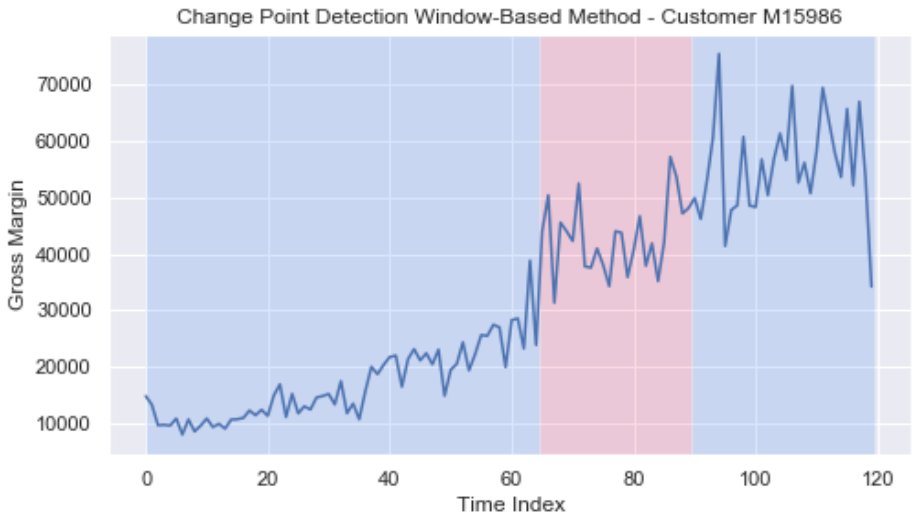

Fig. 11. Another change point detection window

The algorithm requires one parameter that is the window size, that represents the number of samples to calculate the statistical properties to be used on the comparison. The current size window size is set to thirty samples, this could be tuned to improve on the identification of the change point in the time series.

4 Cloud Architecture

An anomaly detection model only works if users are able to consume its results in a timely manner. For a distributed company, building and running a model on a local machine is not a scaleable way of consuming results. XYZ uses a suite of internal tools to manage their operations, sales, and resources. Most of these tools and applications have been migrated to the Cloud. Thus, an anomaly alerting system requires integrating with various data sources and consumers in a Cloud ecosystem.

4.1 Data Ingestion

The ERP (Enterprise Resource Planning) system provides the base data for our model. However, as CRM (Customer Relationship Management) utilization increases it will also become a source of data in our model. We need to be able to export data from both systems to a common location that can be accessed for modeling. It was was determined by our stakeholders that a daily alert would appropriately meet the needs of account managers and the sales team. This means that the model would need to be retrained daily to include new data. This requires a script to run daily that pulls data from sources. This can be achieved

by using a serverless process that establishes direct database connections with internal applications and API calls for third party applications (ex. ERP and CRM). This data can then be written to a temporary file that gets persisted in a document cold store (Fig 11 number 1). All cloud providers offer some form of serverless products that offers pay by usage Cloud computing resources which are useful for handling scheduled tasks. A document cold store is a cheap document storage solution used for archiving and housing data. The document store becomes the one consistent place the model ingests data from.

4.2 Data Modeling

Data modeling (Figure 12 number 2) takes place on its own cloud computing instance. Daily, there would be a process to pull the newest data records from the document store, transform this new data, and append the data records to the training data itself. This new training data gets persisted on the Cloud computing instance, which then only requires new records to go through the memory intensive transformation steps. This data can now be used to retrain the model. The model can then be tested and at-risk clients can be detected. The trained model can be serialized, written to a file, and saved directly on the instance, while the model efficacy metrics (based off test results) and at-risk clients are persisted on a Cloud managed database instance. The database not only provides a history of previous predictions but also an efficient way to query results for consumption.

4.3 Prediction Consumption

Upon saving the prediction results to the database, there has to be a means of notifying consumers that the newest predictions are ready to be consumed. This is done by writing to a cloud hosted message broker. Most message brokers provide a means of partitioning messages for specific consumers [1]. Figure 11 number 3 shows an internal application receiving a message from the broker which triggers an API call to the model instance via an API layer that queries the database. If the hand off between the message broker and internal application breaks down, the broker would have the ability to replay those messages without issue. The consuming application now has the data and can provide an in-application alert that best suits the user workflow. This infrastructure scales for new consumers by adding new partitions to the message broker.

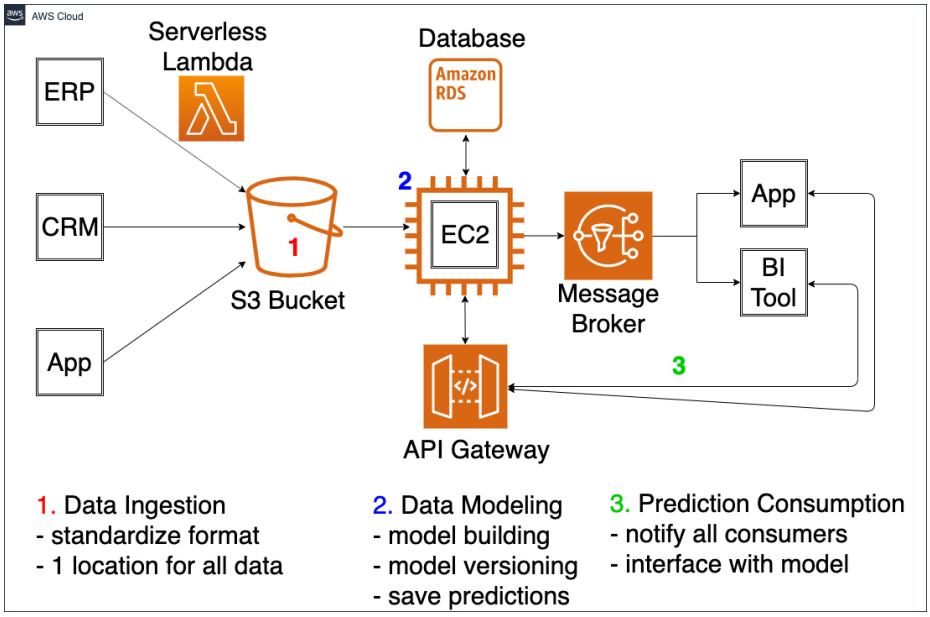

Fig. 12. Cloud infrastructure for data ingestion (1), data modeling (2), prediction consumption (3)

5 Results

From the top 47 customers we observed that 20 customers were at risk and 27 were stable. With this process we were able to identify 13 of the 20 customers resulting in a 65% recall for this model. Though the chance of correct identification is better than a coin toss, there is an opportunity to improve on this metric identify more at risk customers (table 4). The model predicted a total of 16 customers at risk while in reality only 13 demonstrated a drop in gross income metric. This resulted in a precision score of 81%. For the overall model, we were able to flag 37 out of 47 customers correctly, which yields a 78% accuracy for the model (table 5).

Table 4. Top Customer Classification

	Observed	
Predicted	At-Risk	Stable
At-Risk	13	3
Stable	7	24

Table 6 shows the results obtained by the *ruptures* algorithm. The dates in the table represent the most recent change in behaviour for the customers

Table 5. Confusion Matrix

		Observed	
		At-Risk	**Stable**
Predicted	**At-Risk**	TP	FP
		65%	11%
	Stable	FN	TN
		35%	88%

presented in figure 10 and 11. The algorithm provided the braking points for all 47 customers, however for simplicity we are only showing these two. And, as previously mentioned this information can be useful for investigating events that may have contributed to the reduction in gross income

Table 6. Ruptures Last Break Point

Customer	Date	Ruptures Index
M16110	2015-09-01	80
M15986	2016-02-01	85

6 Ethical Considerations

Automated at-risk client identification can help the business get ahead of customer churn and take corrective action. However, nowhere do our predictions prescribe what corrective actions to take. Thus, we need to account for the human impact that could result, especially when it comes to employees at XYZ Packaging.

Most large accounts at XYZ are managed by an account manager that builds the client relationship. If our predictions are used as the sole driver for determining performance then we may run into situations where people are wrongfully terminated or promoted. Our model does not account for external factors such as the client's financial status and product focus. Also there is a degree of error that fundamentally exists. Thus, an account manager that manages a client that pivots product lines may wrongfully be considered under-performing when in fact that person may be very effective at their job. Ethically, it is important to consider that people's livelihoods could wrongfully be impacted if our predictions are used in isolation.

7 Conclusions

The process has a precision of 81% which can help identify at-risk customers and reduce the number of false alarms. This presents an opportunity to improve

customer retention and credibility among the sales team. Of the 20 at-risk customers in the data, 65% of them were correctly classified. This delivers more than a 50% chance of identification which has the potential to reduce customer acquisition costs by reducing customer churn. The 77% accuracy yielded by the model leverages identification of stable customers, which can provide an example of best practices and a measure of customer retention.

References

1. Driesen, V., Eberlein, P.: Brokered cloud computing architecture. U.S. Patent 8,250,135 (2012)
2. Killick, R., F.P., Eckley, I.A.: Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association (107), 1590–1598 (Oct 2012)
3. Truong, C., O.L., Vayatis, N.: Ruptures: change point detection in python. arXiv preprint (1801) (Jan 2018)
4. Zhang, G.: Time series forecasting using a hybrid arima and neural network model. Neurocomputing (50), 159–175 (Jan 2003)