Paper SAS1777-2018

Using SAS® Visual Investigator to Enforce Model Tuning Best Practices in a Regulatory Environment

Scott Wood, Josh Lincoln, and Edwin Rivera, SAS Institute Inc.

ABSTRACT

A common process of model tuning in the Financial Crimes space is sampling data to be submitted to subject matter experts for dispositioning to determine whether activities are suspicious. Establishing these target values is a key component to tuning models and being able to produce more efficient and effective models for catching misuse of the US financial system, as well as identifying terrorist financing activity. In addition, providing traceability of the process to establish thresholds and weights for models is critical for regulated aspects of detecting suspicious activity. SAS® Visual Investigator not only provides a toolkit for data exploration and analysis, it is a great tool for implementing analytics with a workflow process attached to it. This paper describes an approach for using SAS Visual Investigator as an interface for end-to-end model building that incorporates a sampling and disposition process while enforcing a workflow and highlighting the auditability of the process.

INTRODUCTION

Financial institutions (FI) in the United States are required to comply with numerous federal and state regulations related to the prevention of money laundering and detection of terrorism-financing activities. Starting with the Bank Secrecy Act of 1970, certain FIs are required to have a compliance program of people, processes, and technologies in place to detect suspicious activity that might be the result of customers or entities laundering money through the US financial system. The sophistication of the program is generally commensurate to the size and complexity of the organization. Large FIs often employ thousands of employees to assist in monitoring the risk that money launderers pose to their institutions. Big data and data science often form the foundation of an FI's monitoring program because the volume of transactions is impossible for a human to manage without the aid of technology.

Although there are many aspects of a compliance program, including process definition, risk analysis, and training, SAS provides a set of tools and solutions that aid FIs in meeting their regulatory obligations from a quantitative standpoint. FIs often use numerous models for initially detecting suspicious transactional activity, risk-rating customers, and segmenting customer bases. They have more advanced models that can prioritize workloads or suggest optimal ways to monitor customers. With the increase in expectations from regulators and the general adoption of advanced analytical techniques in the compliance space, the process of building an effective and productive model has become a critical topic for many FIs. It is not acceptable to just have a system anymore. The models and rules deployed within the system are under heavy scrutiny to ensure that they are properly built and that weights and thresholds are set in a manner that ensures that the organization is adequately monitoring their stated risk, but conversely is able to staff and support the workload that the models might detect.

Both regulators and internal audit divisions are now highly involved in the evaluation of the creation of new models, initial model training, and subsequent model tuning based on the results and subjective decisions from investigators trained in detecting suspicious activity. Because the decision in determining suspicious activity in the BSA/AML space is a subjective one (as opposed to a fraudulent activity, which is black or white), a common approach to training and tuning is to provide sample alerts to investigators, and have them evaluate the alert and provide a disposition. Plotting the positive target values (productive alerts) against various threshold values enables a data scientist to provide the business unit with a set of threshold parameters related to the model to get the highest number of potential suspicious activity alerts, while not overloading the investigative staff with potential false positive alerts.

The sampling, training, and tuning process is a critical part in an FI's ability to justify its monitoring program. It is critical that all steps are documented to ensure that settings can be justified in the case of a regulatory examination or internal audit. This paper seeks to outline a simple use case using SAS software (SAS Visual Investigator) to load pre-created alerts and to enable the FI to follow a workflow that not only aids users in managing the process, but also provides auditability to the process. Although some users outside of compliance might find this exercise to be heavy relative to its practical value, this level of traceability is an expectation in the AML compliance space related to model risk management (MRM) or audit and many other compliance-related functions. An example is the recent DFS 504 regulations in NY State requiring compliance staff to personally sign off on the viability and performance of their monitoring programs. Although this paper uses an example within the BSA/AML compliance space, the concepts and tools leveraged could be applied to any situation where a high level of scrutiny is placed on the effectiveness of models and the processes and decisions taken to properly tune them.

In this paper, we explore the use of SAS Visual Investigator to load samples of model output to a user interface, follow a simple workflow, and collect the disposition values for subsequent tuning. This paper does not discuss the specifics of any particular model. It explains the operational side of managing the sample of alerts and enforcing an auditable process flow. As such, there are key areas related to using SAS Visual Investigator that are covered.

- Alert sampling and model tuning process
- Loading alerts into SAS Visual Investigator
- Assigning alerts to investigative staff
- Monitoring and auditing the process of collecting disposition data

This paper also outlines some basic concepts and terminology related to SAS Visual Investigator. The key to effectively leveraging any operational system for alert management is having a good sense of the overall process and how it maps to the underlying technology. The use case starts with a simple process flow:

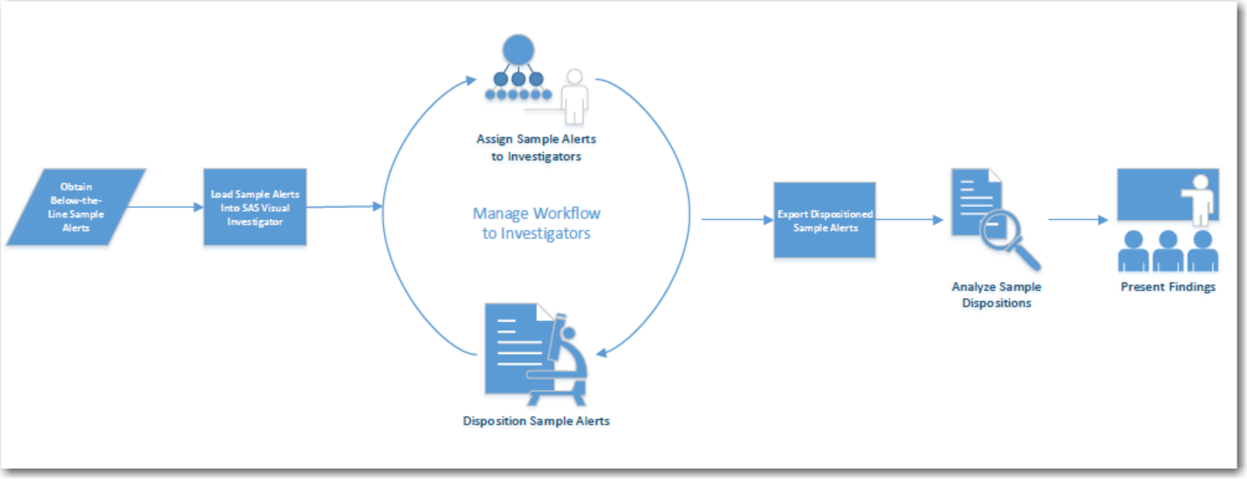

Obtain Below-the-Line Sample Alerts
Load Sample Alerts Into SAS Visual Investigator
Assign Sample Alerts to Investigators
Manage Workflow to Investigators
Disposition Sample Alerts
Export Dispositioned Sample Alerts
Analyze Sample Dispositions
Present Findings

Figure 1. Business Process Flow

ALERT SAMPLING AND MODEL TUNING PROCESS

FIs are responsible for model tuning on a periodic basis. Different FIs have different tuning schedules, but it is common to perform model tuning on a yearly basis. One common approach for tuning anti-money laundering scenarios is to perform Below-the-Line (BTL) testing, which refers to sampling alerts for analyst review that are not currently being generated under the production-tunable threshold values. It uses statistical sampling, which is the selection of a random subset of individual alerts within the BTL testing population for analyst review (with the goal of yielding knowledge regarding the productivity rate of the entire BTL testing population as a whole). Although BTL testing generally refers to alerts that exist below the scenario's minimum tunable threshold values, it also applies to alerts that are not being generated because they are above a threshold ceiling value where applicable. A simple way to view this task is by asking, "What am I missing with the tunable thresholds set to where they are currently?"

BTL testing is used to both detect and estimate the Type II (i.e., false negative) error rates that exist due to the tunable threshold value settings within the AML transaction-monitoring scenarios. Although the regulators—both the OCC and FDIC—are interested in banks reducing their unproductive alert volumes or Type I (i.e., false positive) error rates to increase their monitoring efficiency, their primary concern is ensuring that banks are not failing to generate and investigate suspicious alerts due to their tunable thresholds being set to overly restrictive values. In fact, the regulators generally focus a large portion of their attention on making sure that the Type II (i.e., false negative) error rates are being adequately addressed within the scenario-tuning process via BTL testing of scenario threshold values and back-testing of any analytic alerts models that might be in place to reduce unproductive alert volumes.

The overall sampling methodology is separated into approaches that are used to determine four things:

1. Minimum BTL testing level
2. Number of scenario periods
3. Selection of alerts from the BTL population
4. BTL testing sample size

MINIMUM BTL TESTING LEVEL

Setting the minimum BTL parameter values involves determining how low (or how high in the case of max parameters) the thresholds should be set to calculate the population of alerts eligible for BTL testing. The population of alerts eligible for BTL testing consists of the delta between the production scenario job run and the BTL testing job run for the particular date of interest. The following items should be considered when setting the minimum threshold BTL level:

- The empirical distributions of the various tunable threshold populations for the scenario and where, within such distributions, the current threshold values exist.
- The percentage of the eligible population that is currently generating alerts in production.
- The delta of alerts should be large enough to allow for random sampling.
- The delta of alerts should represent a significant percentage of the total eligible alert population (at least 5%, if not more).
- Whether the tunable threshold acts as a floor, ceiling, or corridor.
- If the scenario logic uses AND or OR logic when generating alerts.
- Where the productive alerts in production reside with regard to their threshold values. (If they are close to the current threshold boundary, then it might be prudent to set a wider BTL testing region and perform enhanced testing.)
- The overall level of productivity for the scenario.
- Whether the same minimum BTL testing level is used for the scenario as a whole or if each segment is set separately.

NUMBER OF SCENARIO PERIODS

Because both the number of alerts and the alert productivity rate differ from run to run, the number of scenario jobs and the periods selected to be included in the BTL alert population need to be determined in advance. This is especially important because reviewing historical alerts can often cause problems within the operational environment and systems. In general, the more frequent the scenario is run, the greater the number of scenario jobs that should be included in the population. (Scenario jobs should be from non-consecutive periods unless all periods are being used.)

There is often considerable alert variability between one scenario job run and the next. This variability might be in the form of different alert volumes, alert attributes, productivity rates (or, in many cases, all three). In addition to general variability from period to period, there might be seasonal factors that impact these items. For these reasons, it is important to use multiple job runs covering a variety of different time periods when creating the population of alerts to sample for BTL testing. (See Figure 2. BTL Testing Sample Process Flow.) Although there is no single correct number of runs to consider, it is important to use enough runs to be comfortable that the BTL alert pool will be representative of the actual total alert population in production.

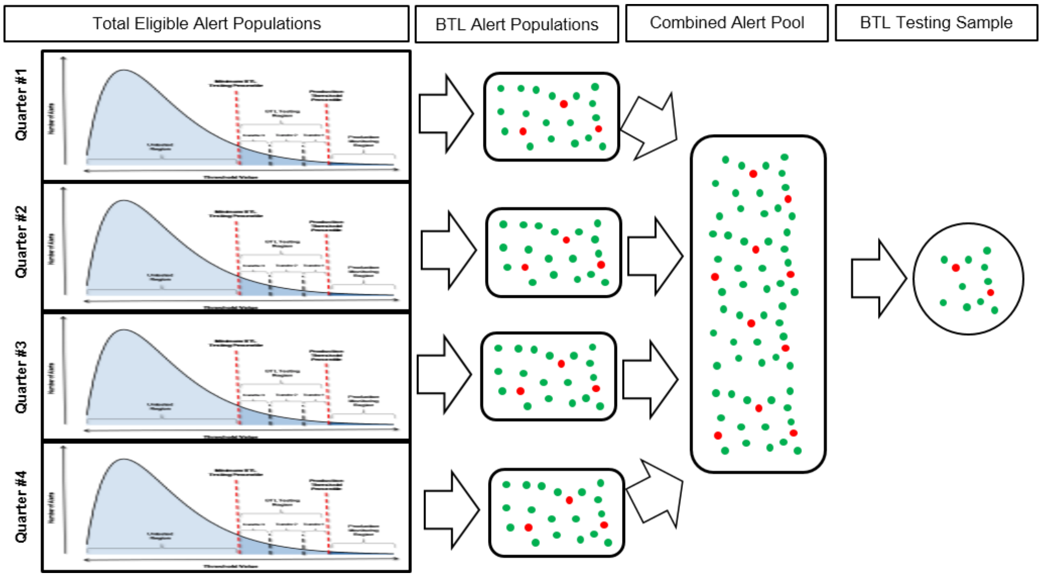


Figure 2. BTL Testing Sample Process Flow

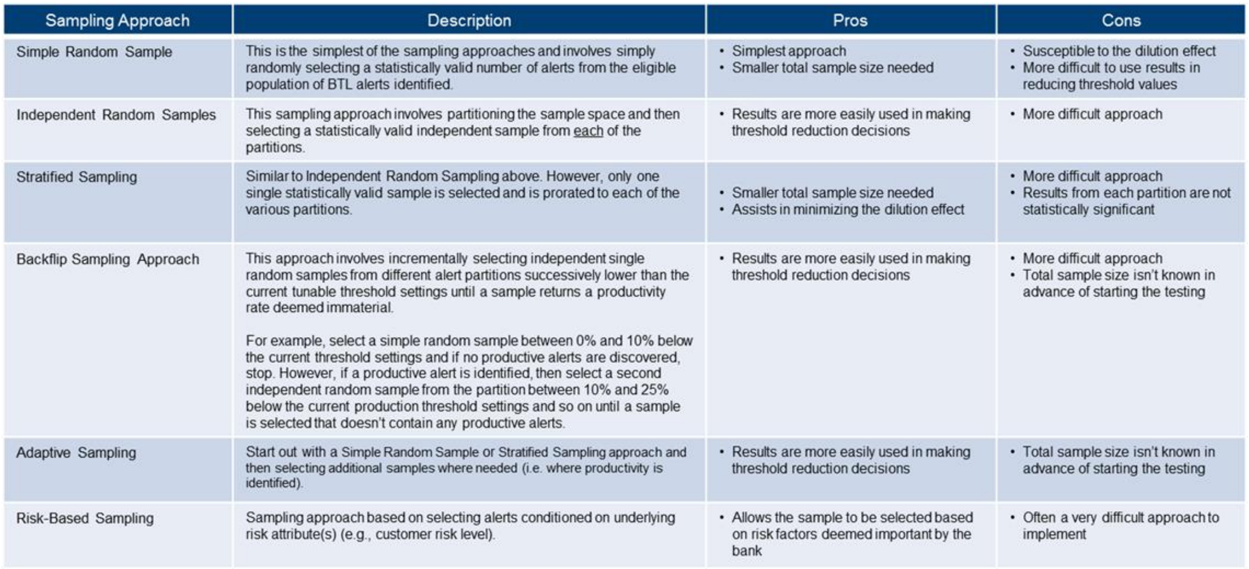
Here are some key considerations:

- Decide which technical environment and data sources should be used from respective transaction-monitoring systems to generate the alerts and how to emulate the models for tuning.
- Remember that alert volumes, tunable threshold distributions, and productivity rates generally vary from period to period, often significantly.
- The greater the number of periods (i.e., run dates) that are included in the BTL testing population, the more likely the population and sample from that population will reflect the actual alert characteristics and productivity rates that existed over the tuning period of review.
- Many regulators think that BTL samples should be selected from at least two separate nonconsecutive months to account for variability.
- Working historical alerts can sometimes cause issues with either the case management system, analyst resource systems, or analyst review procedures (e.g., the customer account is closed).
- Scenarios that generate small volumes of alerts or that are run more frequently might require additional jobs to be run.

SELECTION OF ALERTS FROM THE BTL POPULATION

Once BTL alerts have been generated and production alerts have been removed from the population, what is next? This section covers the approach that should be used to actually select the individual items from the population of alerts. Different sampling methods have different benefits and costs associated with their utilization. Banks need to consider the regulatory expectations of their level of testing, the empirical distribution of their alert population, the granularity for which they want to set thresholds, and their operational constraints when selecting the sampling approach to be used within the tuning process.

Figure 3. Sampling Methods lists some of the more common sampling approaches used to select alerts for AML BTL testing. Although there is an endless number of possible variations that could be used, the five listed cover some of the major types.



Sampling Approach	Description	Pros	Cons
Simple Random Sample	This is the simplest of the sampling approaches and involves simply randomly selecting a statistically valid number of alerts from the eligible population of BTL alerts identified.	• Simplest approach • Smaller total sample size needed	• Susceptible to the dilution effect • More difficult to use results in reducing threshold values
Independent Random Samples	This sampling approach involves partitioning the sample space and then selecting a statistically valid independent sample from each of the partitions.	• Results are more easily used in making threshold reduction decisions	• More difficult approach
Stratified Sampling	Similar to Independent Random Sampling above. However, only one single statistically valid sample is selected and is prorated to each of the various partitions.	• Smaller total sample size needed • Assists in minimizing the dilution effect	• More difficult approach • Results from each partition are not statistically significant
Backflip Sampling Approach	This approach involves incrementally selecting independent single random samples from different alert partitions successively lower than the current tunable threshold settings until a sample returns a productivity rate deemed immaterial. For example, select a simple random sample between 0% and 10% below the current threshold settings and if no productive alerts are discovered, stop. However, if a productive alert is identified, then select a second independent random sample from the partition between 10% and 25% below the current production threshold settings and so on until a sample is selected that doesn't contain any productive alerts.	• Results are more easily used in making threshold reduction decisions	• More difficult approach • Total sample size isn't known in advance of starting the testing
Adaptive Sampling	Start out with a Simple Random Sample or Stratified Sampling approach and then selecting additional samples where needed (i.e. where productivity is identified).	• Results are more easily used in making threshold reduction decisions	• Total sample size isn't known in advance of starting the testing
Risk-Based Sampling	Sampling approach based on selecting alerts conditioned on underlying risk attribute(s) (e.g., customer risk level).	• Allows the sample to be selected based on risk factors deemed important by the bank	• Often a very difficult approach to implement

Figure 3. Sampling Methods

In selecting a sampling method, consider the following factors:

- What are the expectations of the bank regulators?
- The overall operational analyst resources available to conduct BTL testing because some methods result in larger sample sizes.
- The need to know the total BTL sample size before the reviews begin or whether additional BTL alerts can be added after reviews have begun.
- The level of granularity in threshold reductions that the banks want to be able to make once the BTL testing exercise is complete. For high alert volume scenarios, this might be important.
- If dilution of the sample is an issue (impacted by the threshold distribution).
- The level of risk aversion of the bank.
- If there are any risk attributes that the bank feels should be part of the BTL alert sample selection process.

BTL TESTING SAMPLE SIZE

This section covers the appropriate sample size to be selected, including how to infer the population productivity rate after the sample is collected. There are two primary methodologies discussed for determining the sample size to be selected: estimating productivity using a fixed level of precision sample size methodology and binomial exact test sample size methodology. However, there are numerous other approaches available that are not discussed here.

1. Estimating productivity using a fixed level of precision sample size methodology
    a. This approach is based on the normal approximation of the binomial distribution and requires assumptions to be made for the productivity rate, precision, and confidence level.

    b. This is a reasonable approach when the BTL productivity rate is expected to be significantly greater than zero.

c. As the assumed productivity rate decreases, the sample size decreases as well, which is counter-intuitive. (The actual reason for this is based on the variance properties of the binomial distribution.)

d. This approach is really meant to estimate a nonzero BTL productivity rate, but it can be used to perform an approximate hypothesis test as well.

2. Binomial exact test sample size methodology

   a. For the binomial exact test sample size approach, a productivity rate equal to a predetermined cutoff level is assumed. You should derive the sample size that would result in not detecting any productive alerts within that sample to occur only by chance by some small probability.

   b. This approach is used when testing a BTL productivity rate that is expected to be close to zero as it statistically tests this assertion.

The following should be considered when selecting the assumptions for the sample size calculation:

- Will the bank regulators highly scrutinize the assumptions used? (Likely, the answer is yes.)
- Is an overall productivity rate being used for all scenarios or will different scenarios use different assumptions? (This applies to segments as well.)
- The bank's current overall productivity rate and the productivity of the individual scenarios.
- What level of productivity does the bank deem to be immaterial and willing to miss when generating alerts (both the rate and the number of alerts)?
- The operational analyst resources available to conduct BTL testing because more conservative assumptions result in larger sample sizes.
- The level of conservatism that the bank wants to use in its BTL testing process.

LOADING ALERTS INTO SAS VISUAL INVESTIGATOR

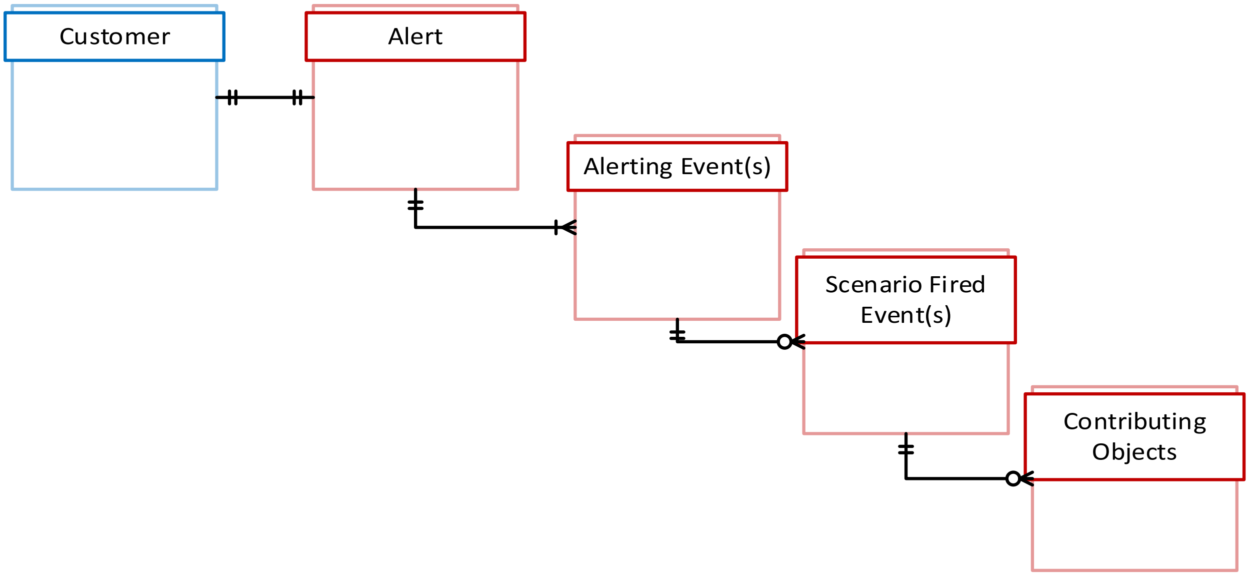
Assuming you have created an appropriate sample of alerts for BLT, there are two options for loading alerts into SAS Visual Investigator:

1. Calling a REST API using JSON syntax
2. Leveraging the Scenario Administrator tool within SAS Visual Investigator

Your specific situation and skill sets will likely drive the approach taken to load alerts. We will cover both options in this paper, but first it is important to gain an understanding of how SAS Visual Investigator defines an alert.

In SAS Visual Investigator terminology, an "alert" is a group of one or more alerting events for the same entity. Each alerting event is the result of a process that indicated the entity should be reviewed. That process often consists of several discrete conditions (rules and models) called "scenarios," which can be passed to SAS Visual Investigator within the alerting event as sub-records called "scenario fired events." This terminology differs from how many investigators and compliance officers refer to alerts in which an entity can have many alerts associated with it, each of which can be dispositioned separately. The same idea exists in SAS Visual Investigator in that an entity can have many alerting events, which are loosely the equivalent of how investigators refer to alerts, but those events are now bundled together into a single object, which SAS Visual Investigator calls an alert.

Because alerts themselves don't represent specific events or conditions, but are containers for the events that require investigation, when an entity has an open alert (when SAS Visual Investigator receives additional alerting events for that entity), the new alerting events are associated with the existing alert. If an alert previously existed on the entity, but has since been completed or closed, the prior alert is reopened as a new work item, and the new alerting events are appended to it until it is again closed.



Given that the intent of model tuning is to compare the efficacy of different models that might generate alerts on the same entity, and considering that SAS Visual Investigator appends alerting events for the same entity to an open alert if one exists, it's important that batches are run only when the alerts from the prior batch have been fully dispositioned. Otherwise, it's impossible to differentiate the dispositions on different batch runs and, therefore, between models.

REST API APPROACH

The SAS platform, built on a micro-service architecture with open access as a top priority, includes numerous REST APIs making ingesting data and interacting with system actions very simple. The SAS Visual Investigator alert micro-service includes an API for receiving JSON-formatted alerting events. Each alerting event *must* include the ID of the entity that's being alerted, and that entity *must* already exist in SAS Visual Investigator. In addition, the JSON for each alert can include the queue to place the alert in, the scenarios (e.g., models or rules) that generated the alert, other entities involved in the alert (e.g., transactions that contributed to the alert), and other attributes beyond what is needed for the model tuning use case.

Submitting the output of the batch run to SAS Visual Investigator is easily achieved by simply creating the necessary JSON structure via PROC JSON, obtaining an access token for SAS Viya using an account with the Create Alerts permission, and then submitting the JSON as a payload using PROC HTTP. Although SAS Visual Investigator can accept nested JSON that more directly represents the relationships between the objects (e.g., nesting scenario fired events within an alerting event), it is likely easier for SAS programmers to use the alternate flattened representation. Here is an example of the flattened representation for a single alerting event:

Alerting Events

Field	Description
alertingEventId	Unique identifier for the alerting event.
actionalEntityType	The type of entity for the actionable entity. Must match an entity type previously configured in SAS Visual Investigator.
actionableEntityId	Identifies the entity that should be investigated.
score	Integer.

Field	Description
alertOriginCd	The process that ran the batch.
alertTypeCd	A string describing the category of alert. E.g. Fraud, Abuse, Manual, etc.
alertTriggerTxt	Explanation of the alerting event, including relevant values from the scenarios.
recQueueId	Identifies a *suggested* queue for the alert.

Scenario Fired Events

Field	Description
alertingEventId*	Identifies the parent alerting event.
scenarioFiredEventId	Unique identifier for the scenario fired event.
scenarioId	Identifies the scenario associated with this scenario fired event.
scenarioFiredEntityType	The type of entity for the actionable entity in the parent alerting event. Must match an entity type previously configured in SAS Visual Investigator.
scenarioFiredEntityId	Identifies the actionable entity in the parent alerting event.
scenarioDescription	Textual description of the associated scenario.
scenarioOriginCd	Indicates the process that executed the scenario.
score	Integer.
displayFlg	Boolean indicating if this scenario fired event should be displayed in the user interface.
displayTypeCd	String indicating the type of visualization to use to represent this scenario fired event. Currently, only TEXT is supported.
recQueueId	Identifies a *suggested* queue for the alert.

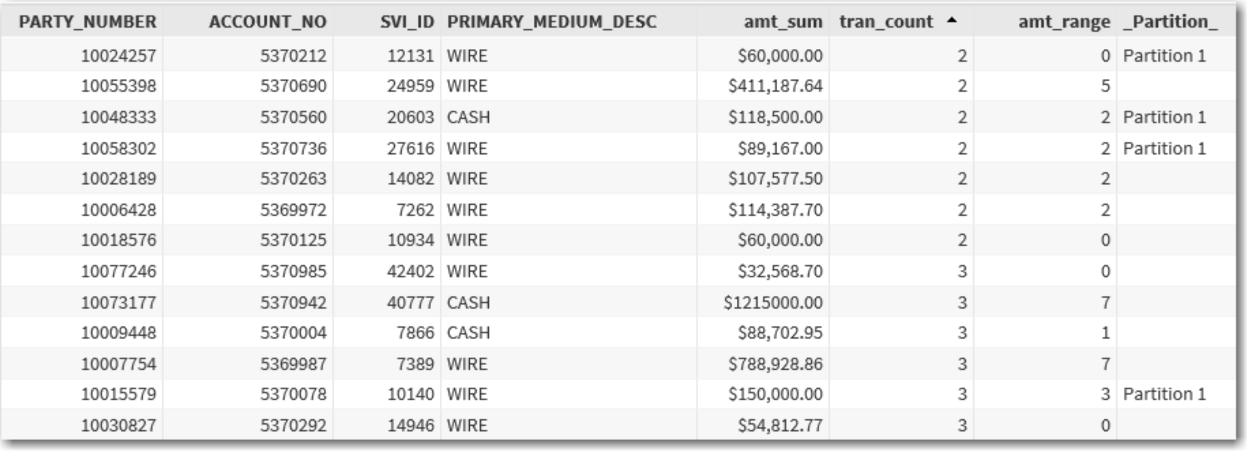
Contributing Objects

Field	Description
alertingEventId*	Identifies the parent alerting event.
scenarioFiredEventId*	Identifies the parent scenario fired event.
contributingObjectType	The type of entity for this object. Must match an entity type previously configured in SAS Visual Investigator.
contributingObjectId	The ID of this instance of the entity, which must already exist in SAS Visual Investigator.
triggeringFlg	Boolean value indicating if this object was a primary factor in causing the scenario to trigger an alerting event.

*These attributes are required only when using the flat JSON structure. When using the nested JSON structure, the parent IDs can be inferred.

SCENARIO ADMINISTRATOR APPROACH

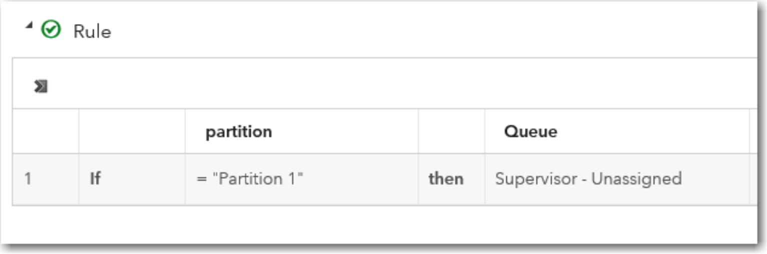
The scenario administrator user interface in SAS Visual Investigator provides another option for loading alerts into the system. In our example, we performed all of our key feature engineering and sampling tasks ahead of time and started with a simple table at an entity or customer level who had alerted based on high aggregate amounts of high risk transactions such as wires or cash. Our alerting dataset was also partitioned based on a stratified sample so that only a small but representative sample of alerts would be sent to the investigation staff.



PARTY_NUMBER	ACCOUNT_NO	SVI_ID	PRIMARY_MEDIUM_DESC	amt_sum	tran_count ▲	amt_range	_Partition_
10024257	5370212	12131	WIRE	$60,000.00	2	0	Partition 1
10055398	5370690	24959	WIRE	$411,187.64	2	5	
10048333	5370560	20603	CASH	$118,500.00	2	2	Partition 1
10058302	5370736	27616	WIRE	$89,167.00	2	2	Partition 1
10028189	5370263	14082	WIRE	$107,577.50	2	2	
10006428	5369972	7262	WIRE	$114,387.70	2	2	
10018576	5370125	10934	WIRE	$60,000.00	2	0	
10077246	5370985	42402	WIRE	$32,568.70	3	0	
10073177	5370942	40777	CASH	$1215000.00	3	7	
10009448	5370004	7866	CASH	$88,702.95	3	1	
10007754	5369987	7389	WIRE	$788,928.86	3	7	
10015579	5370078	10140	WIRE	$150,000.00	3	3	Partition 1
10030827	5370292	14946	WIRE	$54,812.77	3	0	

Figure 4. Alert sample data example

After populating the table above in our database, the metadata for the table was easily imported into SAS Visual Investigator using the administrative user interface and was picked up by the scenario administrator. For our BTL use-case, a record level scenario was chosen which essentially produces one alert per record in the input table that meets certain criteria as configured in the user interface in Figure 5:



Rule

		partition		Queue
1	If	= "Partition 1"	then	Supervisor - Unassigned

Figure 5. Simple rule example

After testing our scenario using the scenario administrator test button and ensuring the expected number of alerts, the scenario and associated flow were published and executed to generate alerts for investigation. These alerts will be sent to subject matter experts trained to detect suspicious activity related to money laundering or terrorist financing.

ASSIGNING ALERTS TO INVESTIGATIVE STAFF

SAS Visual investigator provides a framework for alert management with a user interface for configuration and maintenance. It uses the concepts of Strategies and Queues to handle surfacing alerts to investigators.

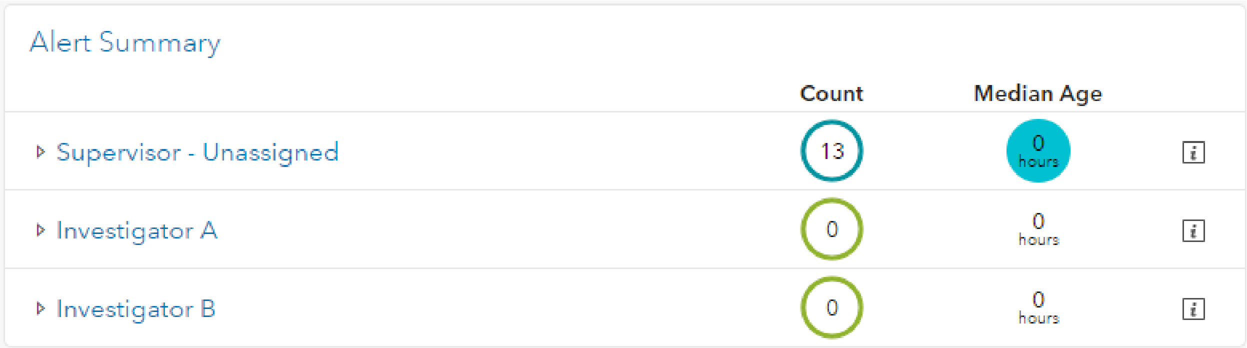
- Strategy – A high level grouping of alerts in which security access can be controlled by group membership

- Queue – A more granular child of a Strategy that can be used to further control who receives alerts and even how subsequent pages displays alerts

For example, all AML compliance alerts may be routed to a single strategy with underlying queues based on the type of product or customer that triggered the alert.

For our sampling exercise we want to route alerts to a supervisor, and subsequently allow the supervisor to assign the alerts out to individual investigators. Three strategies were configured, each corresponding to a specific individual. Each strategy has a single queue configured. Both the scenario administrator and the REST API provide the ability to specify which queue should be used to route the batch of alerts being created. In our case, all alerts will be initially routed to the supervisor's queue.

Figure 6 shows 13 alerts in the supervisor's alert queue.

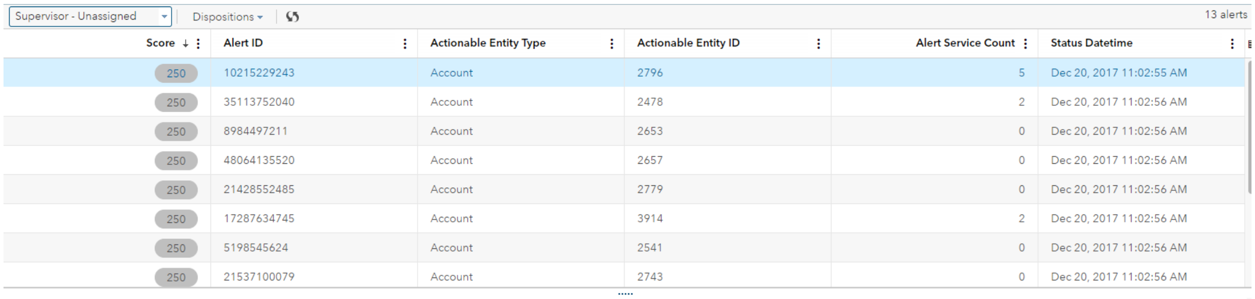


Alert Summary

	Count	Median Age
Supervisor - Unassigned	13	0 hours
Investigator A	0	0 hours
Investigator B	0	0 hours

Figure 6. Supervisor's Alert Queue

Figure 7 shows that by clicking on **Supervisor – Unassigned**, you can see the unassigned alerts in the queue. Alerts can be either individually selected or Shift key-selected for multiple alerts to then be dispositioned or assigned to an investigator.

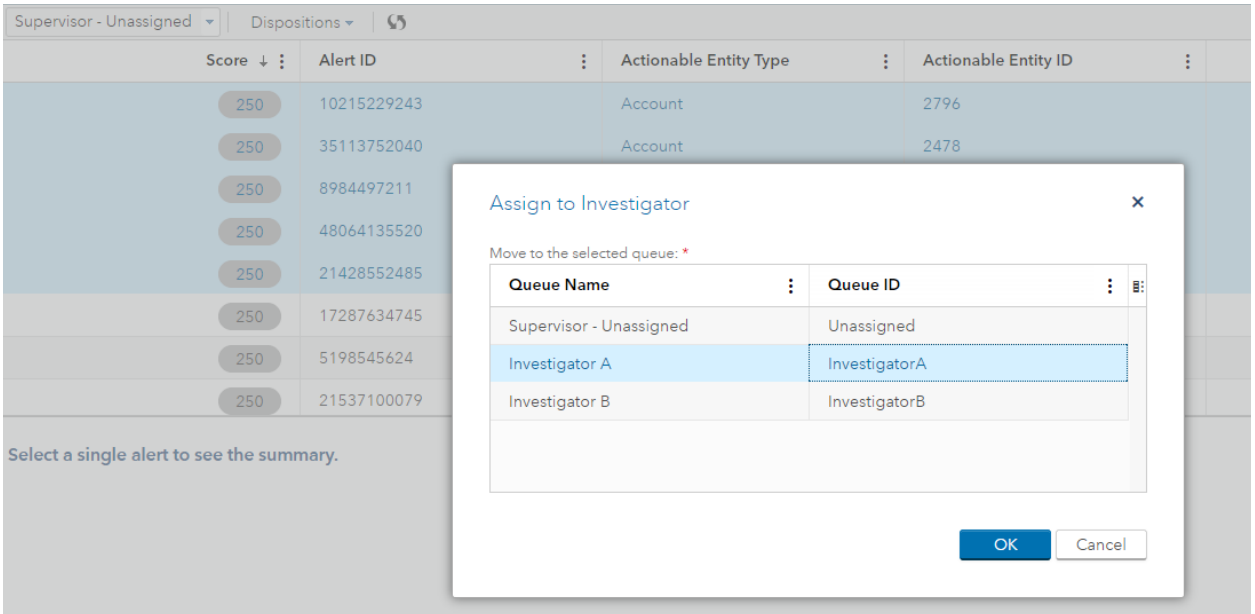


Supervisor - Unassigned   Dispositions   13 alerts

Score	Alert ID	Actionable Entity Type	Actionable Entity ID	Alert Service Count	Status Datetime
250	10215229243	Account	2796	5	Dec 20, 2017 11:02:55 AM
250	35113752040	Account	2478	2	Dec 20, 2017 11:02:56 AM
250	8984497211	Account	2653	0	Dec 20, 2017 11:02:56 AM
250	48064135520	Account	2657	0	Dec 20, 2017 11:02:56 AM
250	21428552485	Account	2779	0	Dec 20, 2017 11:02:56 AM
250	17287634745	Account	3914	2	Dec 20, 2017 11:02:56 AM
250	5198545624	Account	2541	0	Dec 20, 2017 11:02:56 AM
250	21537100079	Account	2743	0	Dec 20, 2017 11:02:56 AM

Figure 7. Supervisor's Alert Queue - Unassigned

Figure 8 demonstrates changing the queue assignment of a group of alerts to an investigator queue and out of the supervisor unassigned queue.



Supervisor - Unassigned | Dispositions

Score	Alert ID	Actionable Entity Type	Actionable Entity ID
250	10215229243	Account	2796
250	35113752040	Account	2478
250	8984497211		
250	48064135520		
250	21428552485		
250	17287634745		
250	5198545624		
250	21537100079		

Select a single alert to see the summary.

Assign to Investigator

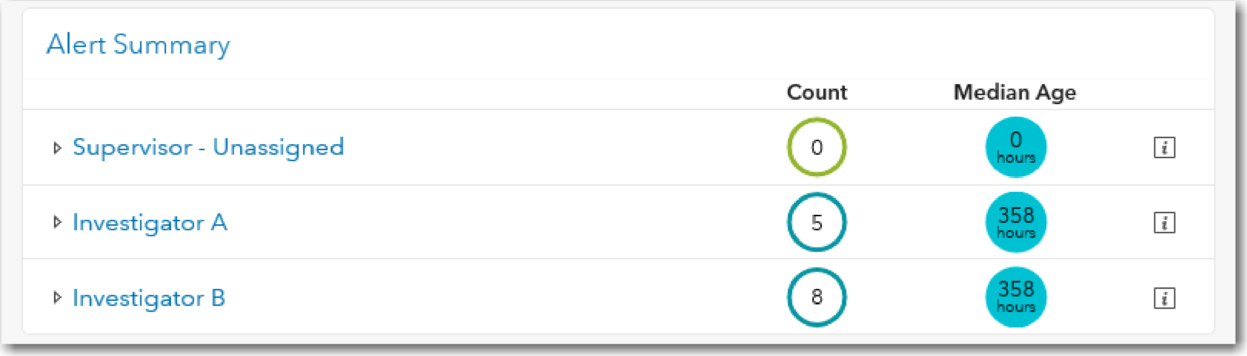
Move to the selected queue: *

Queue Name	Queue ID
Supervisor - Unassigned	Unassigned
Investigator A	InvestigatorA
Investigator B	InvestigatorB

Figure 8. Move alerts to investigators queue

DISPOSITIONING OF ALERTS

After a supervisor has assigned alerts to the various queues, users will see a summary count of alerts per queue that they have access to on their home pages as seen in Figure 9.

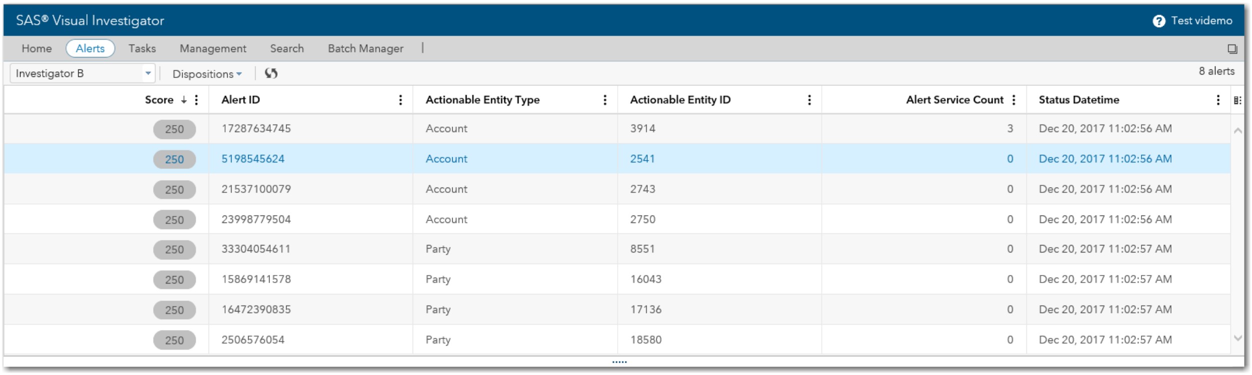


Alert Summary

	Count	Median Age
Supervisor - Unassigned	0	0 hours
Investigator A	5	358 hours
Investigator B	8	358 hours

Figure 9. Alert Summary

Clicking on the name of the strategy takes the user to a list view of the individual alerts.

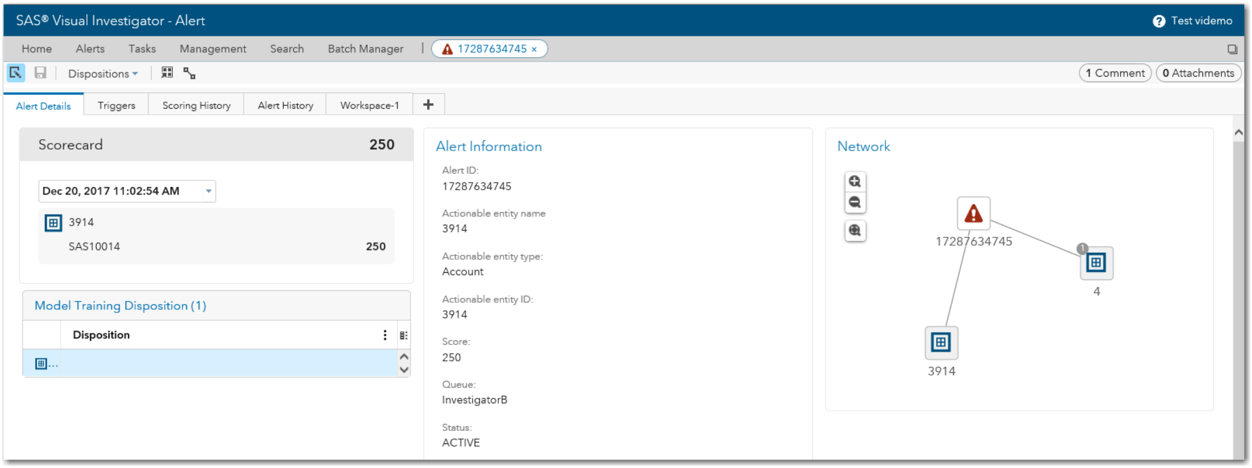


8 alerts

Score	Alert ID	Actionable Entity Type	Actionable Entity ID	Alert Service Count	Status Datetime
250	17287634745	Account	3914	3	Dec 20, 2017 11:02:56 AM
250	5198545624	Account	2541	0	Dec 20, 2017 11:02:56 AM
250	21537100079	Account	2743	0	Dec 20, 2017 11:02:56 AM
250	23998779504	Account	2750	0	Dec 20, 2017 11:02:56 AM
250	33304054611	Party	8551	0	Dec 20, 2017 11:02:57 AM
250	15869141578	Party	16043	0	Dec 20, 2017 11:02:57 AM
250	16472390835	Party	17136	0	Dec 20, 2017 11:02:57 AM
250	2506576054	Party	18580	0	Dec 20, 2017 11:02:57 AM

Figure 10. List View

Clicking on an individual alert takes the user to an interface that presents the detailed information about an alert and enables the user to leverage the investigative functionality of SAS Visual Investigator to review the alert.



Scorecard 250

Dec 20, 2017 11:02:54 AM

3914

SAS10014 250

Model Training Disposition (1)

	Disposition
...	

Alert Information

Alert ID:
17287634745

Actionable entity name
3914

Actionable entity type:
Account

Actionable entity ID:
3914

Score:
250

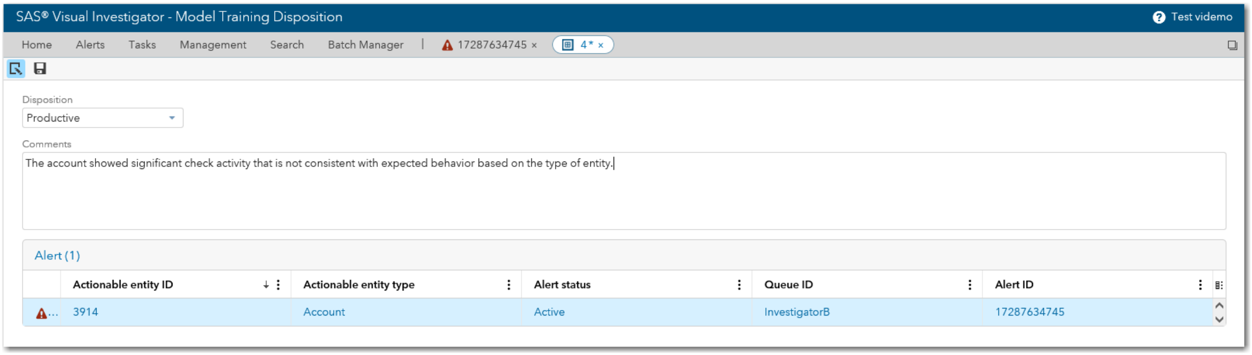
Queue:
InvestigatorB

Status:
ACTIVE

Figure 11. Detailed View

It is from this detailed view that investigators begin to perform their analysis of the entity based on the contents of the operational system, drawing on any prior suspicious behavior or relationships to entities not directly tied to the account. Investigators will also add comments and form narratives, sometimes attaching information pertinent to the decision of whether the alert is suspicious or warrants a deeper investigation. Investigators may seek additional information from other areas of the organization or third parties to augment their investigation and arrive at their final disposition. All of this information can assist in future feature engineering and improving model efficiacy if properly captured and incorporated back into the tuning process.

Setting the disposition on alerts for this project is completed by opening the **Model Training Disposition** object in the bottom left of the detailed alert screen in Figure 11. After opening this object, the user is presented with a screen that enables him to enter the final disposition and any summary-level comments associated with his decision as seen in Figure 12. A custom disposition was created for this project to illustrate the possibility of capturing a variety of factors impacting the disposition of the alert, data points that could allow for more sophisticated and fine grain tuning.



Disposition
Productive

Comments
The account showed significant check activity that is not consistent with expected behavior based on the type of entity.

Alert (1)

	Actionable entity ID	Actionable entity type	Alert status	Queue ID	Alert ID
...	3914	Account	Active	InvestigatorB	17287634745

Figure 12. Disposition and Comments

MONITORING THE PROCESS

Up to this point, the sampled alerts have been loaded and assigned, and the process of analyzing the alerts has been started by various analysts. To manage the completion of the set of sampled alerts, a custom object called “Batch” was created. It brings together a set of sample alerts so that a data scientist or business analyst can monitor the progress of the alerts under review.

The Batch object was set up as an external data object in SAS Visual Investigator. A bridge table was created to link the alerts for a respective batch of sampled alerts. The intention of this custom object is for a user to be able to monitor the progress of the full set of sampled alerts and know when they have all been dispositioned so that the dispositions can be extracted and used for further model tuning.

Pages were created to display these external data objects, and visualizations have been added so that the user can monitor the operational aspects of whether all alerts have been dispositioned and who the alerts are assigned to. Here is the final result:

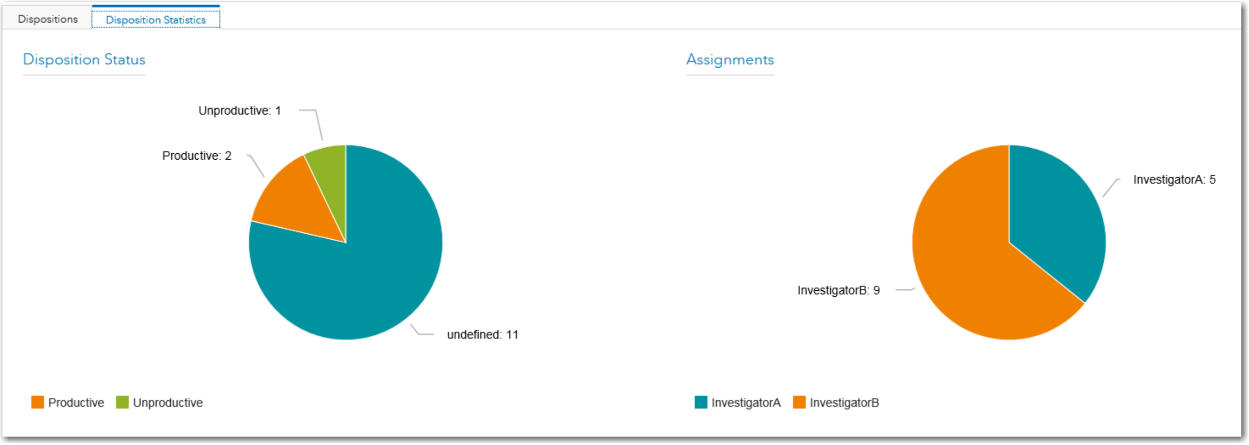
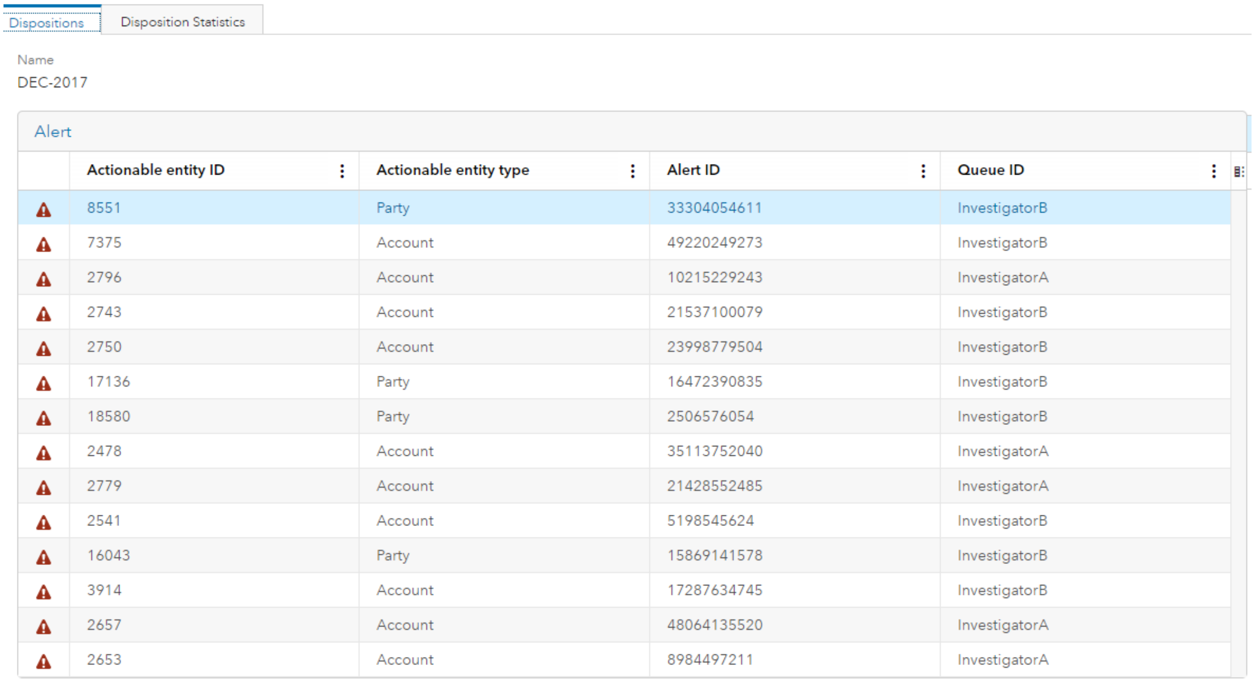


Figure 13. Disposition Status

Figure 14 shows a custom tab to which only the supervisor has permission to view. This allows the supervisor to view the detailed alert assignments in a list view.



Name
DEC-2017

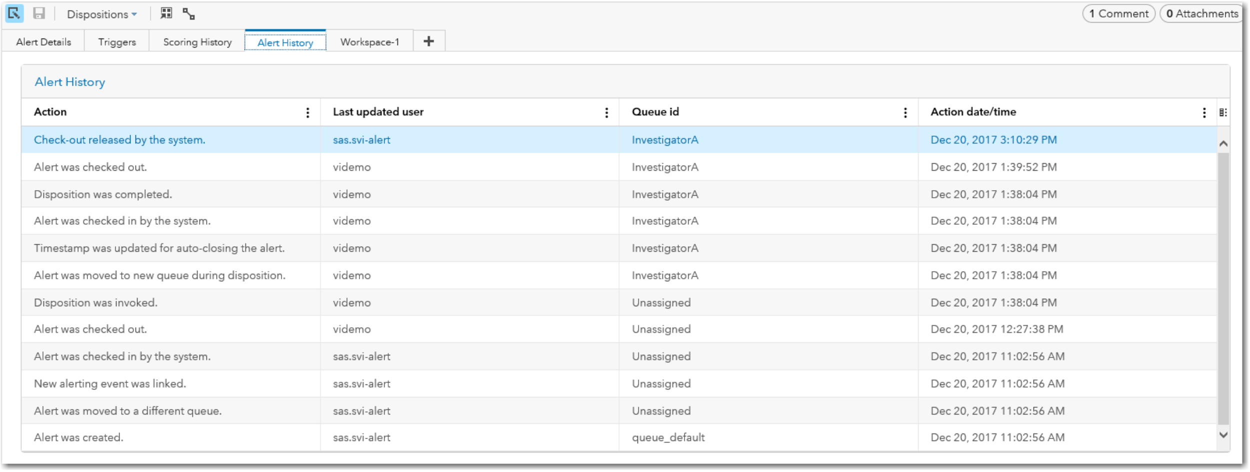
Alert

	Actionable entity ID	Actionable entity type	Alert ID	Queue ID
⚠	8551	Party	33304054611	InvestigatorB
⚠	7375	Account	49220249273	InvestigatorB
⚠	2796	Account	10215229243	InvestigatorA
⚠	2743	Account	21537100079	InvestigatorB
⚠	2750	Account	23998779504	InvestigatorB
⚠	17136	Party	16472390835	InvestigatorB
⚠	18580	Party	2506576054	InvestigatorB
⚠	2478	Account	35113752040	InvestigatorA
⚠	2779	Account	21428552485	InvestigatorA
⚠	2541	Account	5198545624	InvestigatorB
⚠	16043	Party	15869141578	InvestigatorB
⚠	3914	Account	17287634745	InvestigatorB
⚠	2657	Account	48064135520	InvestigatorA
⚠	2653	Account	8984497211	InvestigatorA

Figure 14. Supervisor Only View

REVIEWING THE AUDIT TRAIL

One of the key objectives for leveraging SAS Visual Investigator to manage this process is the traceability and audit that it provides to future users or owners of the tuning process. Not only do the comments, attachments, workspaces, and insights provide information about the disposition of the alert, the **Alert History** tab provides a detailed list of actions taken on the alert from the time of creation through to the current state of the alert. Figure 15 shows a simple example of the lineage of actions taken on a specific alert and the user who performed the respective action.



Action	Last updated user	Queue id	Action date/time
Check-out released by the system.	sas.svi-alert	InvestigatorA	Dec 20, 2017 3:10:29 PM
Alert was checked out.	videmo	InvestigatorA	Dec 20, 2017 1:39:52 PM
Disposition was completed.	videmo	InvestigatorA	Dec 20, 2017 1:38:04 PM
Alert was checked in by the system.	videmo	InvestigatorA	Dec 20, 2017 1:38:04 PM
Timestamp was updated for auto-closing the alert.	videmo	InvestigatorA	Dec 20, 2017 1:38:04 PM
Alert was moved to new queue during disposition.	videmo	InvestigatorA	Dec 20, 2017 1:38:04 PM
Disposition was invoked.	videmo	Unassigned	Dec 20, 2017 1:38:04 PM
Alert was checked out.	videmo	Unassigned	Dec 20, 2017 12:27:38 PM
Alert was checked in by the system.	sas.svi-alert	Unassigned	Dec 20, 2017 11:02:56 AM
New alerting event was linked.	sas.svi-alert	Unassigned	Dec 20, 2017 11:02:56 AM
Alert was moved to a different queue.	sas.svi-alert	Unassigned	Dec 20, 2017 11:02:56 AM
Alert was created.	sas.svi-alert	queue_default	Dec 20, 2017 11:02:56 AM

Figure 15. Alert History

In addition to several user interface based methods of reviewing auditable information, the underlying SAS Visual Investigator database can also be queried to provide a wealth of information and metrics related to operational system usage:

- Model versions, execution history, parameter history
- Alert history, workflow metrics
- User metrics and general user activity logging within the system
- General operational reports for workload management and summary reporting

CONCLUSION

With increased emphasis on model efficacy and tuning from regulatory agencies, SAS Visual Investigator provides a way to merge traditionally manual activities with the rigor that comes with an operational system. Much of the scrutiny placed on compliance organizations is rooted in disparate or manual processes which are highly prone to human mistakes and even loss of knowledge when compliance staff evolves or turns over. By leveraging systems to centralize the storage and lineage of activities related to model tuning, compliance personnel can be sure that a good audit trail of activities resides within the system and not on individuals desktops.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Josh Lincoln
SAS Institute
Josh.Lincoln@sas.com

Scott Wood
SAS Institute
Scott.Wood@sas.com

Edwin Rivera
SAS Institute
Edwin.Rivera@sas.com