Paper SAS6367-2016

Streaming Decisions: How SAS® Puts Streaming Data to Work

Fiona McNeill, David Duling, and Stephen Sparano, SAS Institute Inc.

ABSTRACT

Sensors, devices, social conversation streams, web movement, and all things in the Internet of Things (IoT) are transmitting data at unprecedented volumes and rates. SAS® Event Stream Processing ingests thousands and even hundreds of millions of data events per second, assessing both the content and the value. The benefit to organizations comes from doing something with those results, and eliminating the latencies associated with storing data before analysis happens. This paper bridges the gap. It describes how to use streaming data as a portable, lightweight micro-analytics service for consumption by other applications and systems.

INTRODUCTION

The Internet has created a culture of people conditioned to expect immediate access to information. Mobile networks have created a society reliant on instant communication. The Internet of Things (IoT) is forming a new era, blending these revolutionary technologies, and establishing information access and communication between objects. This provides a seminal opportunity for organizations to realign their services, products, and even identity to an operational environment that responds in real time.

Hopefully by now, the debate of "What is real time versus the right time" is over. For the purpose of this paper, real time corresponds to latency that is so short that events are impacted as they occur. As such, real-time activity is in contrast to the more traditional, offline use of data, where business intelligence and analytics have been used to make both tactical and strategic decisions. Real time is a time-sensitive need and is essential when the decision needs to occur to avoid impending and undesirable threats, or to take advantage of fleeting opportunities.

In order for organizations to operate in real time, some fundamentals are required. Data input must be emitted and received in real time, as it's being generated, such as it is with sensors transmitting object status and health. The data needs to be assessed in real time, extracting the inherent meaning from the data elements as they are being ingested. Lastly, the data needs to provide decisions and the instructions for low latency actions. These are characteristics associated with streaming data.

Unlike other types of data, streaming data is transferred at high-speed, on the order of hundreds, thousands, and even millions of events per second – and at a consistent rate (save for interrupted transmissions associated with network outages). Popular types of streaming data include streaming television broadcasting and financial market data. Such data are continuous, dynamic events that flow across a sufficient bandwidth and are so fast that there is no humanly perceived time lag between one event and the next. Given the high volume and high velocity of streaming data, it's not surprising that the receipt, ingestion, and decisions made from this data are left to powerful computing technology, which can scale to assure the high-volume, low latency actions by objects connected in the IoT enabling them to communicate and respond in real time.

UNDERSTANDING DATA, ANALYTICS, AND DECISIONS IN IOT

Streaming data is not only high volume, high velocity data, it is also highly varied data. It can be generated by humans, machines, objects, sensors, and devices. With such different sources it should be of no surprise than that streaming data varies in data type, format, specification parameters, and communication protocols.

EVENT STREAM DATA

For simplicity, we can broadly divide streaming data into two major types:

- Structured data, such as continuous readings from heart rate and blood pressure monitors, use tracking from smart meters, binary readings of on/off status from machinery, RFID tags, sensors readings of temperature and pressure from oil drills, banking transaction systems, and more.

- Semi-structured or unstructured, such as data that is generated by computer machine logs, social media streams, weather alert bulletins, live camera feeds, and operational and ERP systems (free form notes and comments are included with the structured records in most operational/ERP systems), to name a few.

It's often assumed that streaming data, generated from sensors, devices and machines is consistent and accurate unlike human generated content, which is known to be fraught with misspellings, stylistic differences, translation loss, and so on. However, sensor data, and its cohorts also suffer from inconsistent and incorrect data, with bad readings (temperature sensor goes awry), missed readings (with interruptions in transmission) and consolidating different readings, which are typically associated with multi-sensor assessment that have different specifications or protocols. For example, dialysis machines communicate using different languages, transmitting in USB, Ethernet, and different serial interfaces (RS-232, RS-485, RS-422, and so on, and Wi-Fi®.

Streaming data, as with any other type of data suffers from data quality issues that must be addressed in order to assess, analyze and action it. Big data repositories, like Hadoop®, provide a currently popular answer to capture and then cleanse streaming data for analysis. And while initially, this might be viable, it's only a short-term stop gap. With the expansion of IoT, and corresponding explosion in streaming data on the horizon even low cost commodity storage for big data will soon be prohibitive to economically address the needs of streaming data. Yet even if an unlimited budget existed (it doesn't), when real-time answers are demanded, the latency associated with first storing streaming data, then cleansing, then analyzing adds incremental time to processing – delaying actions until they are no longer in real time.

You can reduce the transmission, storage, and assessment costs of streaming data by cleansing and analyzing streaming data near the source of the data generation, pushing required processing to the edges of the IoT. Aggregators, gateways, and controllers are natural levees to cleanse multiple sources of aggregated data, minimizing the downstream pollution with dirty events. Embeddable technology, provided by SAS® Event Stream Processing, aggregates, cleanses, normalizes, and filters streaming data while it is in motion – and before it stored. SAS Event Stream Processing is poised to even process data at the sensor processing chip itself.

Unlike traditional database management systems, which are designed for static data in conventional stores, and even big data repositories, with queries to file systems, streaming data management requires flexible query processing in which the query itself is not performed once or in batch, but is permanently installed to be executed continuously. SAS includes pre-built data quality routines in SAS Event Stream Processing query definition. In this way, the necessary streaming data correction, cleansing and filtering is applied to data in motion and in turn, reduces polluting data lakes with bad and irrelevant data. Of equal, if not more importance, including streaming data quality paves the way for streaming analytics, doing the required data preparation for analytically sound real-time actions.

STREAMING ANALYTICS

Tom Davenport, thought leader in field of analytics, has said that the "Analytics of Things is more important than the Internet of Things" (Davenport, 2015). Arguable perhaps by those in the communications industry, the point is that understanding the data upon which connected objects communicate is critical to having successful conversations between the 'things'. IoT provides an opportunity to reconsider how we use analytics, and make it pervasive - to drive useful and effective conversations between things.

In many cases, we can apply the same types of analytics to streaming data that we use in traditional batch model execution. The difference is that unlike traditional analysis, which requires data to be stored before it's analyzed with event streams analyze data before it's stored. The following types of analytics are applicable to IoT data as part of the continuous query:

- Descriptive analytics identifies patterns in events as they occur

- Predictive analytics identifies future likelihoods of events that have not yet happened
- Prescriptive analytics provides the instructions for event actions

SAS Event Stream Processing provides procedural windows to include both descriptive and predictive algorithms defined in SAS DATA step, SAS DS2, and other languages. As with traditional analysis, these models are built in SAS® High-Performance Data Mining, SAS® Factory Miner, SAS® Contextual Analysis, SAS® Forecast Server, and any other SAS® product or solution that generates SAS DATA step or SAS DS2 code. For this broad selection of algorithms, the models are built upon an event history that has been stored. As with traditional analysis, models are built, tested, and validated. The resulting model code, however, is included into the SAS Event Stream Processing continuous query as a pre-defined, procedural calculation. As part of the continuous query, the model scores individual events are they are ingested. In other words, the analytics are performed on live data streams, synonymous with the term 'streaming analytics'.

Taking advantage of machine learning and deep learning techniques, SAS Event Stream Processing includes a growing suite of methods to build and score event data solely based on streaming data, and without out-of-stream model development and event history. In this case, algorithms such as K-Means clustering, are both defined and applied to events in motion, learning with new events. This exciting field of new techniques further expands the streaming analytics methods available to streaming data.

STREAMING DECISIONS

The focus of this paper is enabling prescriptive analytics in stream, a term we describe as 'streaming decisions'. Streaming decisions define the instructions for real-time actions based on live, streaming events. They are of particular importance to actions taken by object in the IoT. They combine descriptive and predictive algorithms, with the business rules that trigger when the models are relevant to the current streaming event data scenarios. In other words, they are the instructions needed by an IoT object to take the right action, something of core importance to the adoption and successful proliferation of autonomous IoT activity.

As we distribute analytics further out to the edges of the IoT there is a classification that provides some guiding principles that direct when one type of analysis, and the corresponding actions, is more applicable than another. The following types of analytics are described on the IoTHub (2016):

- Edge Analytics is the analysis at the same device from which it is streaming
- In Stream Analytics is the analysis that occurs as data streams from one device to another, or from multiple sensors to an aggregation point
- At Rest Analytics is the analysis processed after the event fact has passed, based on saved historical event data and/or other stored information.

In general, the closer to the edge, the less event data there is to analyze. At the edge, there is just that one object/sensor/device, with its limited supply of data. As mentioned, data quality issues are present at the edge, and events can be aggregated in windows of time to correct and filter out the irrelevant noise from the signal of interest. Analytical calculations are more limited due to the data restrictions, and prescriptive analytics (say, instructions emanating from another object) are limited to real-time actions that can be performed in isolation – like commands to turn up or down, turn on or off.

As more objects are related to each other in-stream, at aggregation points, the data are richer (emanating from several sources) and correspondingly there are more data quality issues. The contextual understanding of the scenario is also richer (with more event data over time, space, and so on) and, in turn, more complex patterns of interest can be identified. Streaming decisions can thus be made that relate to more objects, even becoming a series of inter-connected actions.

As typical of any analysis, the decision to apply different types of models is dependent on the data as well as the business problem the analysis solves. More often than not, IoT analytic solutions require multiphase analytics, that is models defined in the traditional, stored data paradigm, and scoring for new analytical insight, as well as in-stream model derivation/calculation, and analytics applied to the edge. SAS® does this. With the same integrated code, and with over one-hundred and fifty (at last count) adapters and connectors linking streaming data, SAS Event Stream Processing is used to define the complete continuous query that can be as simple or complex as the business problem itself. Moreover, built into SAS Event Stream processing is the ability to automatically issue alerts and notifications for real-time situational awareness and understanding of event status.

When we consider the IoT we are describing, an analytically driven network of objects that communicate with each other. When we automate actions between objects, especially when there is no human intervention, the risks associated with rogue actions, as well as the technical debt that accumulates from both machine learning algorithms (Sculley et al., 2015) and from any unmanaged advanced analytics environments, will outweigh the advantages. As such, the IoT demands a governed, reliable and secure environment for streaming analytics and associated prescribed analytic actions. SAS® Decision Manager is a prescriptive analytics solution. With fully traceable workflows, versioning and audit trails to assure command control over streaming analytics for real-time, reliable, and accurate IoT applications.

BUILDING STREAMING DECISIONS WITH SAS®

Decision Construction

When designing, building, and testing decisions it’s best to begin with a description of what is included within a decision. For the discussion within this paper, we consider the types of decision-making that organizations use, that is strategic and tactical decisions, and the ways that organizations leverage analytical models and their output to make decisions that meet businesses goals.

Strategic and Tactical Decisions

Organizations are required to address decisions at both the strategic and the tactical levels since both are required for a business to run effectively. Strategic decisions typically represent the less common decisions that an organization makes such as creating new product lines or expanding into new territories or merging with another firm. These decisions, while important, are not typically made on a frequent basis and therefore businesses can take the time and effort needed to create specific processes that aren’t required to be repeatable nor require automation.

Tactical decisions, on the other hand, and which can include operational decisions, are made frequently and often (often thousands of decisions), made in a single day or even in minutes or seconds. Loan underwriting, fleet maintenance operations, point of sale operations, fraud detection, and remediation are examples of the decisions that process high volumes of rapidly moving information. Tactical decisions like these are numerous, require short timeframes, high rates of data ingestion, as well as automation and analytics, and of course ways to prescribe an appropriate action based on the analytic model output.

Analytics, an important element for tactical decision making, has become pervasive within organizations due to a couple of factors. First, the accessibility of analytics has increased due to the rise of tools that assist and guide users through the analytical process to suggest relevant algorithms based on the available data. This data-driven, guided approach includes better visualizations to identify patterns in the data and recommendations, as to which is the best model to use. Second, analytics is being applied to a wider set of problems, ranging across industries from retail to manufacturing, to health care and drug development. Organizations have come to recognize that the application of analytics can help with a vast array of problems. The emergence of the data scientist and the citizen data scientist represents the wider number of users that are using the new and powerful analytical tools, applying them in a variety of ways to solve difficult and complex business problems within a single business.

SCALING TO DATA STREAMS

Prescribing Action from Analytics

Analytics has become pervasive in business, and can result in large volumes of analytical output, which applied to a business process in order to generate a decision. The output is usually a score, a numerical representation of the likelihood of an outcome. But how can businesses translate that numerical output into a decision that is repeatable, automated, and scalable?

Some scoring models deliver probabilities of an event occurring based on analysis of historical trends and events. This output is derived using, in many cases, IF-THEN-ELSE (conditional) logic applied to models is part of the probability determination, but that score doesn't define the action to be taken. For example, a typical credit score for a customer applying for a loan doesn't specify if the loan officer (or underwriting system) should approve or deny the loan, what rate to offer the customer, or what product to cross-sell/upsell. Comparatively, the actions of a decision would provide the prescriptive instruction suggesting approve over deny, defining the appropriate rate, or that the customer would likely be interested personal insurance with that loan. These prescriptive instructions combine the output of the analytical model along with business rules, suggesting, or defining the action that is needed. Such prescriptive action drives the numerous tactical and operational decisions for a business, and be scalable to address the high-frequency data inherent to business operations.

Decisions in Action

Streaming data, as we have seen, represents data from sources such as customers web clicks, call data records, fleet vehicle GPS, point-of-sale systems, and now more commonly, sensors from corporate assets, such as machines on manufacturing floor, or sensors in an electric power grid.

Data from these systems has historically been extracted, transformed or cleansed from the source, and then loaded to data warehouses for storage and later to analyzed. But, as described above, soon, if not already – organizations will conclude that they can't afford to store it all, and it certainly can't afford the lag times of analyzing it after data has been stored. In business operations, the value of data diminishes the longer we wait to use it, so we need new ways to analyze it sooner – closer to where it originates, and that means we need new ways to tap into the value of data streams.

Tapping into data, while it's still in-motion in data streams, before it's stored empowers actions to be applied sooner, before its value diminishes and before missed an opportunity or prevented a threat. The diminishing value of data can be seen in Figure 1, depicts the relationship between our ability to ingest, and analyze the data for an action and the value of making that decision sooner rather than later.

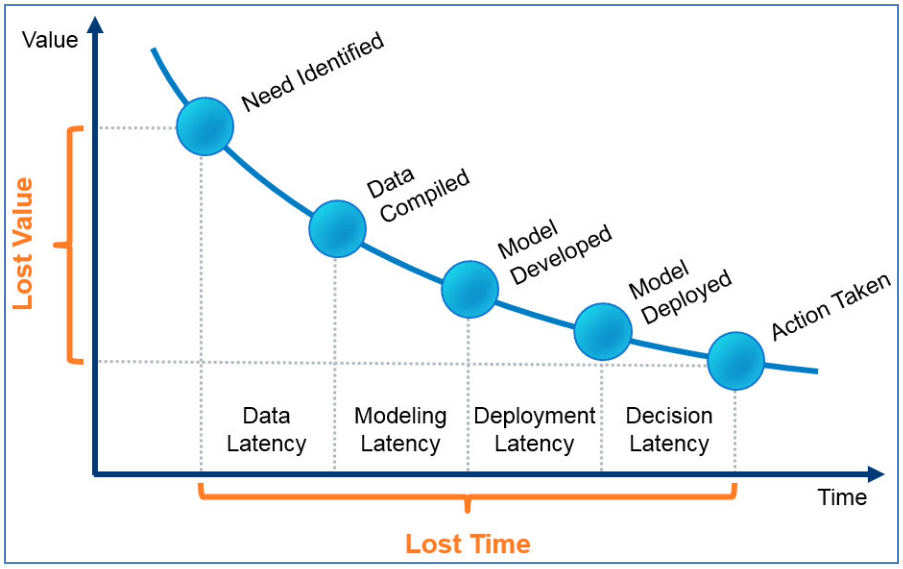


Figure 1. Decision Decay and Diminishing Value

DRIVING INTERNET OF THINGS (IOT) ACTIONS

In this nascent hyper-connected world of IoT, data is being generated rapidly and businesses want effective approaches to leverage their analytical resources to not only analyze and gain insights from the rising tide of data but also take action from it - to obtain the most, real-time value.

IDC research have found that only 0.5% of the data being generated through the IoT is being analyzed to derive value (see Figure 2). This means that only 0.5% of the data from "things" was being analyzed at that time, leaving a rich set of opportunities to understand and take action untapped. And while this research is from 2012, and more organizations have begun to examine IoT data for deriving business value, the vast majority of organizations have not yet used IoT data for business operations. This report also points out the amount of untagged data (often associated with unstructured text data) that would be more useful if it was tagged and analysis-ready. However, with the aforementioned traditional technique of storing first and then analyzing, tagging content for use if often prohibitive because of the consequential large storage costs – even if we assume that all potentially useful unstructured text could be stored (Gantz and Reinsel, 2012).

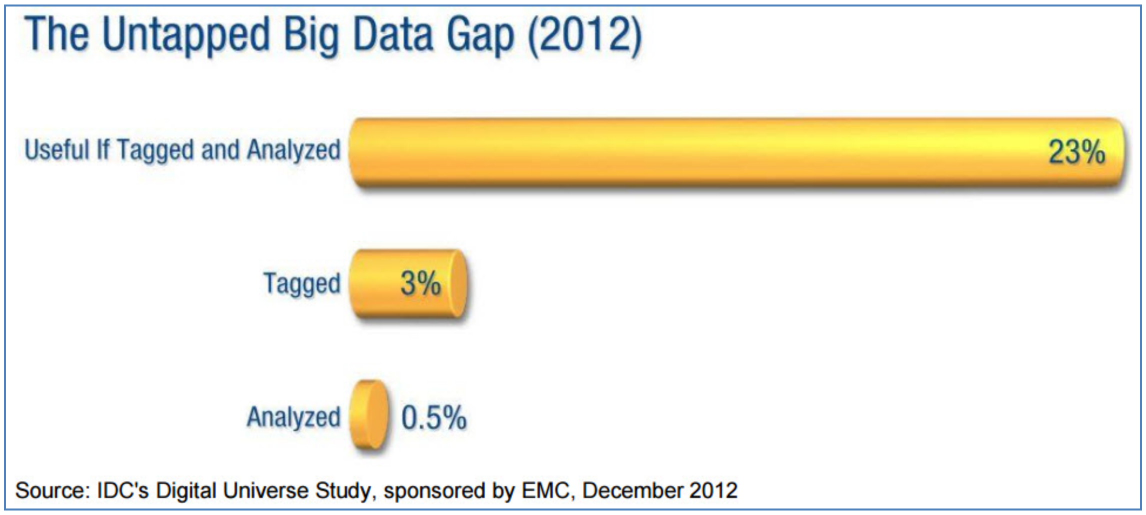


Figure 2. IDC: The Untapped Big Data Gap (Gantz and Reinsel, 2012)

With increasing numbers of objects, sensors and devices joining the connected network of the IoT, the big data gap is growing. Processing data at the necessary scale and speed associated with streaming data generated from the 'things' requires new architectures can analyze and make decisions on the streaming, in-motion data, filtering out the irrelevant from any downstream activity – including data storage. As such, there is a growing necessity to move analytics, the associated operational decisions, and the corresponding actions closer to the data - and in some cases, right into the data streams, near the point of data generation.

SAS Event Stream Processing and SAS® Decision Manager together enable organizations to analyze data while it's still in motion and also apply prescriptive actions sooner, deriving maximum value from live events and before streaming data value diminishes. The communication between prescriptive instruction from SAS Decision Manager and the real-time analytical determination embedded within SAS Event Stream Processing is achieved using SAS® Micro Analytic Services[1]. SAS Micro Analytic Services provides the ability to quickly execute decisions based on the results of in-stream scoring.

[1] SAS Micro Analytic Services are included within the SAS Decision Manager offering.

SAS Decision Manager and SAS Event Stream Processing

SAS Decision Manager is being used to automate tactical decision making by prescribing actions to take through the design, development, testing, and publishing of decision flows. SAS Decision Manager supports the Business Analyst, Data Scientist, Data Miner, and Statistician, collaboratively developing tactical decision actions from building decision flows that combine the analytical models with necessary business rules that drive operational business processes.

These same decision flows, authored from SAS Decision Manager and published into SAS Micro Analytic Services, can be can be executed within streaming data by inclusion in the event data flow activity defined in SAS® Event Stream Processing Studio.

CASE STUDY: PRESCRIPTIVE ACTION FOR TRANSPORTATION

The following real-world case study illustrates the power of streaming analytics calculated close to event data origination for real-time prescriptive actions. We describe a simple yet powerful approach with SAS Event Stream Processing to build the streaming model and analyze streaming data, with SAS Decision Manager acting on the real-time events using SAS Micro Analytics Services.

BUSINESS PROBLEM

A fleet management company wants to minimize the time vehicles are out of service to reduce costs, minimize lost revenue, and maximize uptime. The trucks have sensors that transmit data that monitoring location, vibration, rpm, temperature, speed, steering angle, pressures, and so on. The company needs to proactively prioritize maintenance before a vehicle unexpectedly is out-of-service. The somewhat obvious and immediate business benefit for analyzing vehicle sensors in the transportation industry is to maximize assets efficiently.

Streaming Analytics for IoT Actions: Two Aspects Driving One Outcome

The company has collected data on its fleet and processing it offline, identifying problems, and then generating notices sent to maintenance locations, to try to improve the maintenance scheduling and overall efficiency of vehicle assets. This approach, however, is typically expensive, given the vehicle is often inoperable, taken off the road before parts are available for the necessary maintenance, mechanics can fit the additional work their schedules. This can result in unplanned downtime, overtime servicing costs, expensive non-scheduled parts delivery, and more.

In this scenario, the company wants to use real-time data they're collecting from vehicles to identify issues sooner and increase the time available to address maintenance proactively. In fact, the truck sensors can be analyzed in an onboard SAS Event Stream Processing engine, reducing the latency from the time the data is collected to the time the issue is identified and addressed.

On-board processing of streaming data is one solution to better predict the likelihood of a vehicle issue. An even better solution is to drive a prescriptive action that instructs where to route the vehicle to, notifies suppliers of necessary parts for on-time delivery, and identifies mechanic schedules aligned to a service stop that minimizes deleterious impact of the fleet's transportation of goods to its customers.

Together, the in-stream analysis along with the prescriptive actions enable alerts to be generated in-real time to all stakeholders, empowering them to take the right action, minimizing costs, maximizing revenue and drastically reducing unplanned down times for the fleet.

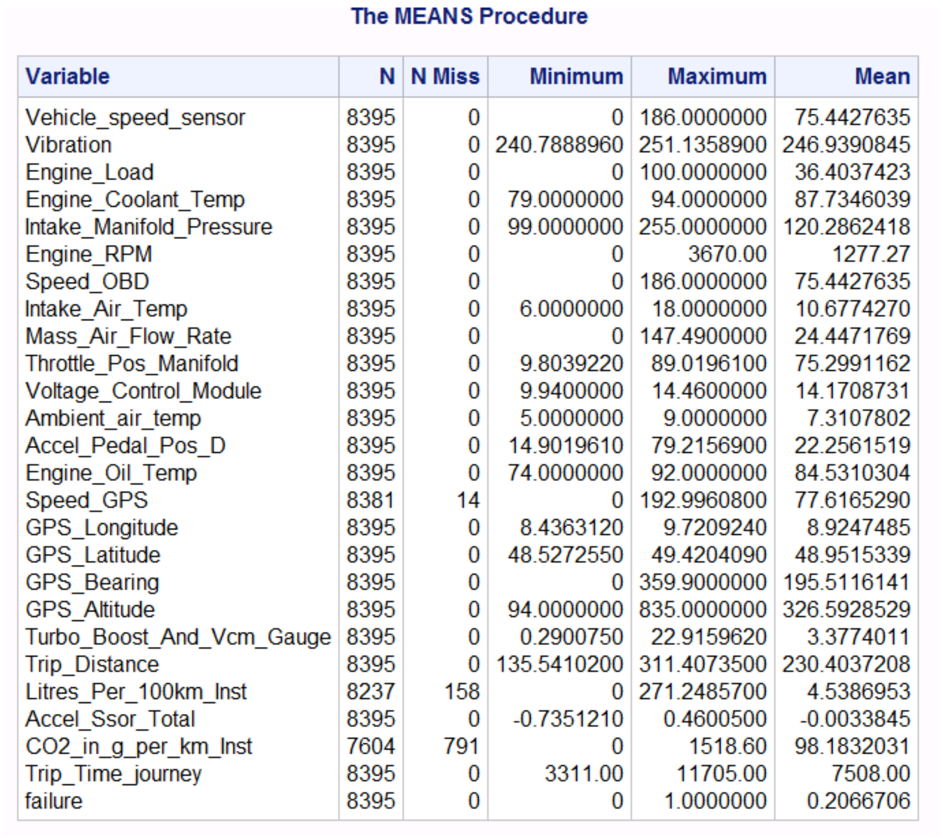
SAS® DECISION SPECIFICATION

Analytics can be found at the core of any quality decision making process. In this case, we want to build a predictive model that will help us decide when trucks are likely to experience a failure that requires downtime and maintenance. We need a history of truck sensor readings and a history of scheduled and unscheduled maintenance events. We can then use the sensor readings to predict the unscheduled events and use those predictions to make better decisions. To build the history data, we have accumulated data from a small fleet of trucks equipped with sensors. These trucks were operated for a period of time in different locations and conditions and their sensor readings stored in on board memory. The saved readings were later downloaded to a database. In addition, we have data on the maintenance records for those same trucks and significant events were labeled as failures. We have joined the sensor readings with the maintenance events to create a predictive modeling table. The vehicle data represents real sensor readings and has been provided by the Intel ™ Corporation for public demonstrations.

Building the Model

The first thing that a data scientist will do is look at the data and run a basic variables distribution analysis. All variables in this sample including *failure* are numeric.

```
proc means data = trucks.failures n nmiss min max mean ;
   output out=means ;
   var _numeric_;
run;
```



The MEANS Procedure

Variable	N	N Miss	Minimum	Maximum	Mean
Vehicle_speed_sensor	8395	0	0	186.0000000	75.4427635
Vibration	8395	0	240.7888960	251.1358900	246.9390845
Engine_Load	8395	0	0	100.0000000	36.4037423
Engine_Coolant_Temp	8395	0	79.0000000	94.0000000	87.7346039
Intake_Manifold_Pressure	8395	0	99.0000000	255.0000000	120.2862418
Engine_RPM	8395	0	0	3670.00	1277.27
Speed_OBD	8395	0	0	186.0000000	75.4427635
Intake_Air_Temp	8395	0	6.0000000	18.0000000	10.6774270
Mass_Air_Flow_Rate	8395	0	0	147.4900000	24.4471769
Throttle_Pos_Manifold	8395	0	9.8039220	89.0196100	75.2991162
Voltage_Control_Module	8395	0	9.9400000	14.4600000	14.1708731
Ambient_air_temp	8395	0	5.0000000	9.0000000	7.3107802
Accel_Pedal_Pos_D	8395	0	14.9019610	79.2156900	22.2561519
Engine_Oil_Temp	8395	0	74.0000000	92.0000000	84.5310304
Speed_GPS	8381	14	0	192.9960800	77.6165290
GPS_Longitude	8395	0	8.4363120	9.7209240	8.9247485
GPS_Latitude	8395	0	48.5272550	49.4204090	48.9515339
GPS_Bearing	8395	0	0	359.9000000	195.5116141
GPS_Altitude	8395	0	94.0000000	835.0000000	326.5928529
Turbo_Boost_And_Vcm_Gauge	8395	0	0.2900750	22.9159620	3.3774011
Trip_Distance	8395	0	135.5410200	311.4073500	230.4037208
Litres_Per_100km_Inst	8237	158	0	271.2485700	4.5386953
Accel_Ssor_Total	8395	0	-0.7351210	0.4600500	-0.0033845
CO2_in_g_per_km_Inst	7604	791	0	1518.60	98.1832031
Trip_Time_journey	8395	0	3311.00	11705.00	7508.00
failure	8395	0	0	1.0000000	0.2066706

Output 1. Output from the MEANS Procedure

We can see that three variables have missing values in a relatively small number of cases: 14, 158 and 791, out of a total of 8395 cases. We can also see that *failure* occurred in 20% of the cases. Therefore, predicting failure has potential to improve efficiency and reduce costs.

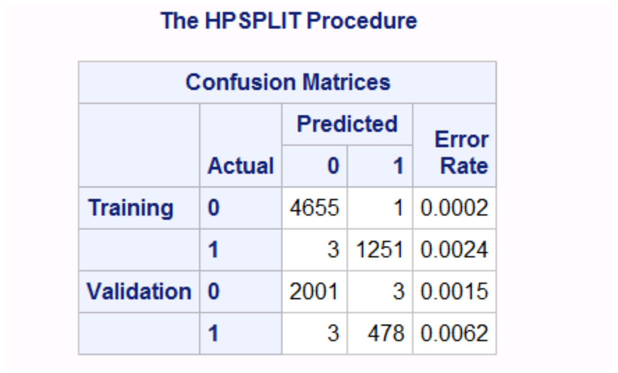
We need a tool that can predict failures for this sample. We need to the tool that has robust handling of missing values. We also need a tool that can provide root causes analysis and determine which devices with sensors potentially contribute to failure so that we can improve those devices and reduce overall failure rate. Finally, and perhaps most importantly, we need a tool that can produce a failure scoring function that we can deploy to the real-time system. The scores generated by this system will be used to generate signals for the decision processing application. Higher scores will indicate that failure is possibly imminent and the truck should be routed to a service center before a serious problem occurs.

Decision tree is a modern data mining tool that handles missing values and is useful for both model interpretation and scoring. A decision tree is sometimes referred to as a recursive partitioning algorithm, which works by selecting variables that have the greatest power to divide cases based on the dependent target variable values. The result is a downward tree where each node contains fewer cases and the dependent target variable distribution is more biased. The prediction value at each leaf of the tree is the leaf's proportion of dependent target variable events. In our case, the target variable *failure* has two values, zero and one, where one represents an observed failure event. We want to create a decision tree model that predicts the event value (one) and identifies the independent input variables that are used in that prediction.

Therefore, we will proceed with building a model using the HPSPLIT procedure from the SAS Enterprise Miner ™ distribution. Notably, we are using the procedure option missing=branch to enable tree branches based on missing values in addition to real values. We also did not select the GPS variables since we want our models to be based on truck sensor readings that can be applied in any location. For the complete procedure syntax refer to SAS documentation.

```
/* select list of input variables */
proc transpose data=means (drop= _TYPE_ _FREQ_ ) out=vars;
       id  _STAT_ ;
run ;
proc sql noprint ;
       select _name_ into :vars separated by ' ' from vars
       where  _name_ ne 'failure' and
              _name_ not contains "GPS";
quit ;
/* predict truck failure */
filename scrcode "&file.\score.sas" ;
proc hpsplit data= trucks.failures missing=branch ;
       performance details ;
       partition fraction (validate=0.3) ;
       input &vars / level=int ;
       target failure / level=nom order=ascending ;
       score out= model ;
       code file= scrcode ;
run ;
```

The results of the HPSPLIT procedure provide some clues about the failure analysis. First we want to examine the accuracy of the procedure. The confusion matrix shown below displays the number of cases that are correctly and incorrectly predicted. In our sample, only six cases of failure were miss-identified as non-failures. The accuracy of this model is very good.



The HPSPLIT Procedure

Confusion Matrices

		Predicted		Error
	Actual	0	1	Rate
Training	0	4655	1	0.0002
	1	3	1251	0.0024
Validation	0	2001	3	0.0015
	1	3	478	0.0062

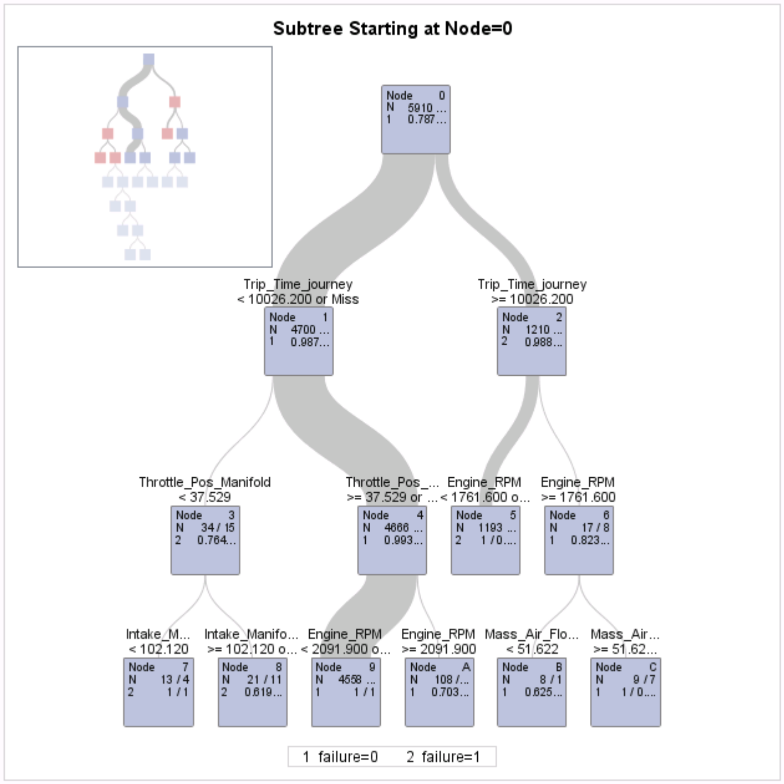
Output 2. Output from the HPSPLIT Procedure

For interpretation of the model, we can look at the list of variables selected as important predictors. The devices attached to these sensors should be examined for their possible contribution to truck failures. These are the variables that will be required to execute the decision scoring function.

Variable Importance						
	Training		Validation		Relative Ratio	Count
Variable	Relative	Importance	Relative	Importance		
Trip_Time_journey	1.0000	42.8206	1.0000	26.9451	1.0000	1
Trip_Distance	0.1533	6.5649	0.1552	4.1811	1.0121	1
Engine_RPM	0.1500	6.4233	0.1441	3.8821	0.9605	2
Throttle_Pos_Manifold	0.1454	6.2267	0.0712	1.9198	0.4900	1
Intake_Manifold_Pressure	0.0357	1.5266	0.0657	1.7690	1.8415	1
Vehicle_speed_sensor	0.0234	1.0000	0.0455	1.2247	1.9463	1
Engine_Load	0.0433	1.8556	0.0329	0.8867	0.7594	2
Mass_Air_Flow_Rate	0.0255	1.0914	0.0205	0.5525	0.8045	1
Accel_Pedal_Pos_D	0.0452	1.9365	0.0000	0	0.0000	1
Litres_Per_100km_Inst	0.0319	1.3644	0.0000	0	0.0000	1

Output 3. Variables Selected by the HPSPLIT Procedure

Finally, we can look at the structure of the decision tree to better understand the complexity of the model and the order of importance of the input variables. The following tree graph shows the overall size of the tree model in the overview box and the readable detail of the top portion of the tree. We can see that *Trip_Time_journey* is the most important predictor, followed by *Throttle_Pos_Manifold* and *Engine_RPM*.



Output 4. Decision Tree Structure from the HPSLIT Procedure

Deploying the Model

Now that we have confidence in this model, we can work on deploying it to the event stream processing engine. Model scoring is simply the application of the model to new data to produce a score. In this case the model represents the decision tree created in the model building step. In the running of the HPSPLIT procedure, we saved the scoring code to an external file named *scrcode* that contains simple DATA step code that generates predictions. The score code contains 159 source code lines of SAS DATA step code that can be inserted between the SET statement and the RUN statement in a SAS program. A small fragment of the score code is displayed below. The score code contains several similar fragments.

```
. . .
IF NOT MISSING(Trip_Time_journey) AND ((Trip_Time_journey >= 10026.2))
 THEN DO;
  IF NOT MISSING(Engine_RPM) AND ((Engine_RPM >= 1761.6))
   THEN DO;
    IF NOT MISSING(Mass_Air_Flow_Rate) AND ((Mass_Air_Flow_Rate < 51.6215))
     THEN DO;
      IF NOT MISSING(Accel_Pedal_Pos_D) AND ((Accel_Pedal_Pos_D >= 27.12156951))
       THEN DO;
        _Node_ = 18;
        _Leaf_ = 8;
        P_failure0 = 0;
        P_failure1 = 1;
      END;
      ELSE DO;
        _Node_ = 17;
        _Leaf_ = 7;
        P_failure0 = 1;
        P_failure1 = 0;
      END;
  END;
  ELSE DO;
    _Node_ = 12;
    _Leaf_ = 3;
    P_failure0 = 1;
    P_failure1 = 0;
  END;
END;
. . .
```

The code requires the input variables listed by the model description and generates five new variables that describe the model. *_WARN_* is an indicator that the prediction function could not be computed from the values of the input variables. *_NODE_* and *_LEAF_* are internal variables that identify the branches taken for each case. *P_Failure1 and P_Failure0* are the probabilities of truck failure and non-failure, respectively. We are primarily interested in *P_Failure1* and *_WARN_.* Higher values of *P_Failure1* indicate that action should be taken to prevent truck failure. Non-empty values of _WARN_ might indicate that one or more critical sensors have failed and action should be scheduled to repair those devices. The score code file is stored as part of the sample code for this paper. Two additional variables for validation data proportion have been omitted as they are not needed for this example. The DATA Step score code is then adapted for the run-time environment.

Event Stream Processing

The score code is now deployed to the SAS Event Stream Processing engine. The SAS Event Stream Processing engine requires code in the DS2 format. DS2 code is a modular and structured form of SAS DATA Step that can be embedded in various run-time environments. The process for converting acceptable DATA Step code to DS2 is fairly simple. The scorexml macro will detect the necessary input variables and save them to an XML file. The DSTRANS procedure will convert the DATA Step code to DS2 code and add the needed input variable declarations. We then want to test the scoring in a simple DS2 procedure step and examine the output.

```
/* Create an XML file for Variable Info */
filename scrxml "&file.\score.xml";
%AAMODEL;
%scorexml(Codefile=scrcode,data=trucks.failures,XMLFile=scrxml);

/* Convert Code to DS2 */
```

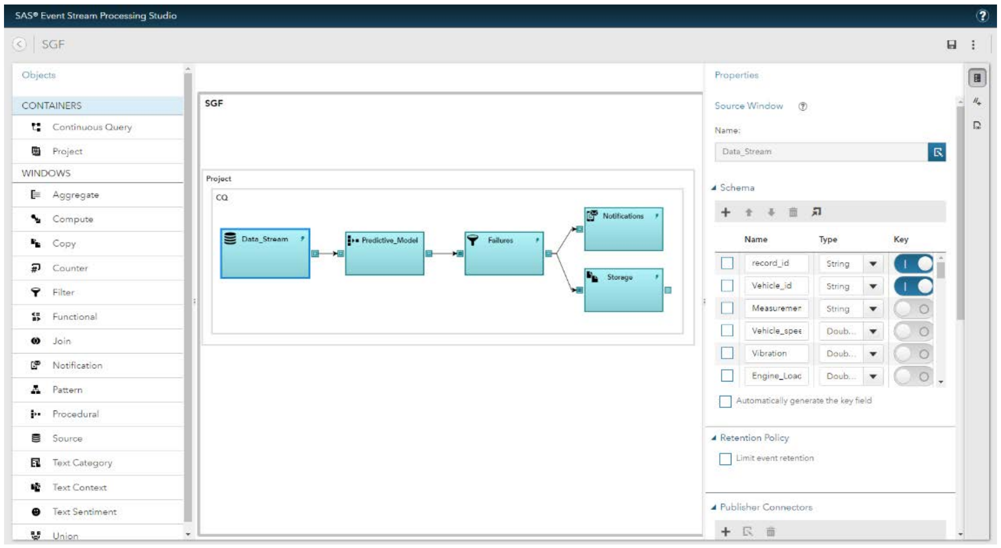
```
proc dstrans ds_to_ds2 in="&file.\score.sas" out="&file.\score.ds2" EP nocomp
xml=scrxml; run; quit;

/* Test Scoring */
libname SASEP "&file.";
proc delete data=sasep.out; run;
proc ds2 ;
      %include "&file.\score.ds2";
run;
```

After we have validated the DS2 code and the scoring, an additional manual step is required to rename the input and output destinations to ESP.IN and ESP.OUT, respectively.

The authors recommend using SAS Event Stream Processing Studio for creating the stream definition. In that definition we need to define a procedural window based on DS2 code. The properties for this window will include a text editor for DS2 code. The user needs to paste the DS2 code created in the previous section into this procedural window. The event definition must include the input variables required by the decision tree model and pass them to the procedural window. The window definition must add the *_WARN_* and *P_Failure1* variables to the output event definition.

Display 1 shows a simple design for handling the sensor events. An on board diagnostic device has already aggregated the sensor data into a single data record. The SAS Event Stream Processing Studio Data_Stream window can subscribe to the diagnostic data records, execute the predictive model scoring, and then filter the events to the ones that have a high probability of failure. After the filter, we can add a notification window that will post predictive failure events to a remote service such as a data center, or store them locally until they can be retrieved by a physical location such as a maintenance shop.



Name: Data_Stream

Name	Type	Key
record_id	String	On
Vehicle_id	String	On
Measuremen	String	Off
Vehicle_spee	Doub...	Off
Vibration	Doub...	Off
Engine_Loac	Doub...	Off

Automatically generate the key field

Limit event retention

Display 1. Simple Design for Handling Sensor Events in SAS Event Stream Processing Studio

The Complete System

The complete solution for monitoring devices on a fleet of trucks, predicting failure, and taking action is very complex. This paper covers only a small part of that system. SAS provides key components for adding advanced analytics to every step of that journey.

SAS Event Stream Processing can integrate high speed data management, pattern detection, and model scoring in integrated devices. Local events can be handled and filtered before integration with operational servers. We used a decision tree scoring function to predict possible failures. As a result, only the important events that indicate failure need to be transmitted to the operational systems. By moving critical analytics to the local system we can detect problems earlier and save huge amounts of server processing, storage, and networking.

SAS Decision Manager can be used to construct and execute routine business decisions in both online and batch processing environments. After an alert from a truck is received, the custom decision logic can be used to determine the best action to take and the best way to execute that action. The core SAS Decision Manager contains components that will help manage and monitor the predictive models, build and manage business rules, and build and execute decision processes. These components operate in a data center making routine business decisions and have business-friendly interfaces. They provide a buffer between SAS Advanced Analytics and the business's operational systems.

The complete system is depicted in Figure 4. The IoT systems are responsible for event processing and pattern detection. Operational systems are responsible for delivering the business strategy and resources to the IoT systems and managing the alerts and communications that are recommended. This view is entirely representational. There are many ways to architect the system and many ways to connect the components.

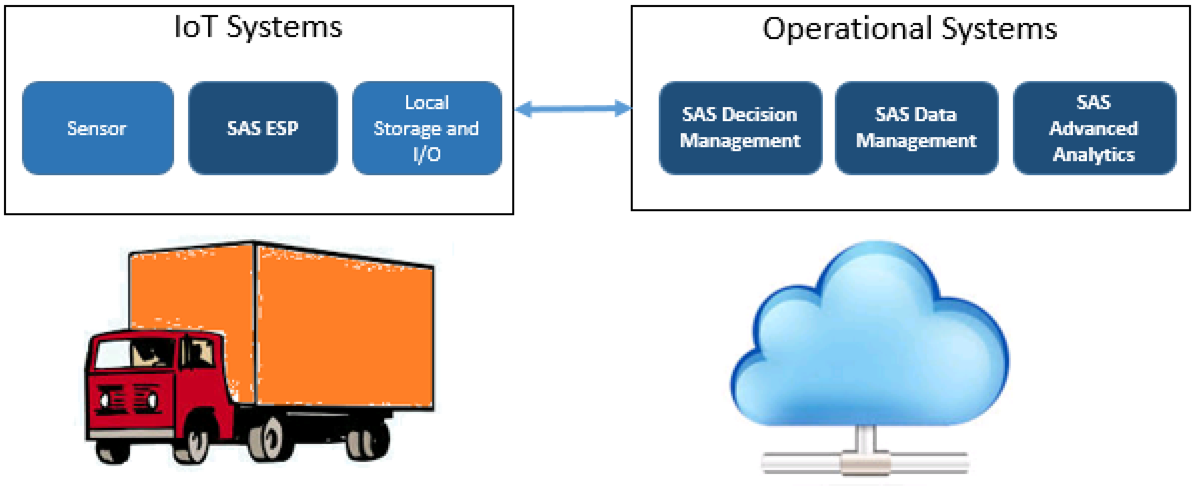


Figure 3. Architecture of IoT Systems and Operations Systems

RESULTS FROM STREAMING DECISIONS

Businesses are required now to adapt to the needs of real-time data analysis and decision making to deliver value as lower costs, rapid response, and maximized revenue. With SAS Decision Manager and SAS Event Stream Processing, they can now deliver that value to the business, minimize the additional costs associated with missing opportunities, and reduce the inherent latencies of processing data after it has been persisted in various data stores. Tangible results include reducing maintenance costs through a more proactive approach and targeting specific maintenance tasks to keep their fleet moving as efficiently as possible. And since time is money, the business can address problems before they manifest themselves as out-of-service vehicles that have slowed delivery times, added costs, and negatively impacted customer satisfaction.

With real-time analysis and decision making applied across their fleet, the business can look at other ways to take advantage of the big data gap in their operations. By looking to new sources of event data, along with those that already exist, and applying streaming analytics, the business could improve inventory turnover (through integration of point of sale data with inventory management), or reduce stock of maintenance components (by integrating live data into their forecasting mechanisms). The possibilities are almost limitless when organizations look for ways to leverage streaming data and improve their decision making.

CONCLUSION

The following summarizes the four ways that processing streaming data provides a vital role in IoT (Combaneyre, 2015):

- Detect events of interest and trigger appropriate action
- Aggregate information for monitoring
- Sensor data cleansing and validation
- Real-time predictive and optimized operations

SAS Event Stream processing is being used by organizations to send an alert, a notification trigger, for current situational monitoring and improving sensor data quality in real time, across industries.

The autonomous actions of objects needs to be founded in well-established practices of humans. As objects in the IoT are depended on to make decisions, the need to govern, manage, control, and secure the conditional logic specific to specific event scenarios will become increasingly important. For these more complex, real-time streaming decisions, the processing speed of SAS Event Stream Processing executing actions defined with the rigor of SAS Decision Management will ensure that real-time actions of autonomous objects in the IoT are both the right and the relevant ones.

REFERENCES

Combaneyre, F. 2015. “Understanding data streams in IoT.” *SAS White Paper.* Available at http://www.sas.com/en_us/whitepapers/understanding-data-streams-in-iot-107491.html.

Davenport, T. 2015. “#ThinkChat IoT and AoT with @ShawnRog and @Tdav” *Information Management.* Available at http://en.community.dell.com/techcenter/information-management/b/weblog/archive/2015/06/22/thinkchat-iot-and-aot-with-shawnrog-and-tdav.

Gantz, J.; Reinsel D. 2012. “The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East.” *IDC IVIEW.*

“IoT pushes limits of analytics”. IoTHub. February 29, 2016. Available at http://www.iothub.com.au/news/iot-pushes-limits-of-analytics-415787.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, D., Crespo, J-F., Dennison, D. 2015. “Hidden Technical Debt in Machine Learning Systems” *Proceedings of NIPS 2015.* Montreal, PQ. Available at http://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf?imm_mid=0df22b&cmp=em-data-na-na-newsltr_20160120.

ACKNOWLEDGMENTS

The authors would like to thank Kristen Aponte, Brad Klenz, and Dan Zaratsian for their assistance and contributions to this paper.

RECOMMENDED READING

“Channeling Streams of Data for Competitive Advantage”, *SAS White Paper* *http://www.sas.com/en_us/whitepapers/channeling-data-streams-107736.html*

“How Streaming Analytics Enables Real-Time Decisions”, *SAS White Paper* http://www.sas.com/en_us/whitepapers/streaming-analytics-enables-real-time-decisions-107716.html

CONTACT INFORMATION

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

Fiona McNeill
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
Email: fiona.mcneill@sas.com

David Duling
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
Email: david.duling@sas.com

Steve Sparano
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
Email: steve.sparano@sas.com