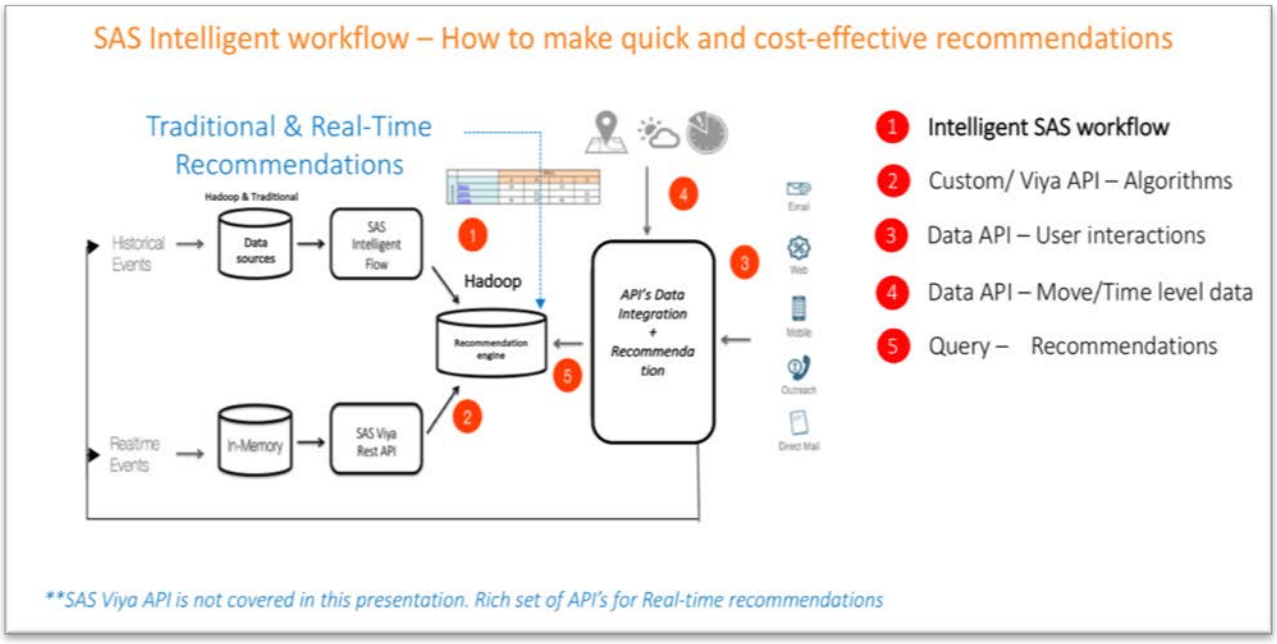
Paper 2401-2018

Deploying and Maintaining Models in a Big Data Environment: An Intelligent SAS® Workflow

Anvesh Reddy Minukuri, Comcast Corporation; Ramcharan Kakarla, Comcast Corporation;
Sundar Krishnan, Comcast Corporation

ABSTRACT

Creating predictive models is one element of the data mining. Implementation and maintenance are another ones. Mostly, We will have two different kinds of data which is real-time and historical data. Historical data is given priority over real-time because of its huge volume and past proven performance. Moreover, real-time needs to predict/capture the data within range of milliseconds and make recommendations using huge-computations and quick Algorithm API's. This paper focuses on the historical data model deployments and maintenance to store the models on Hadoop. With the advent of big data ecosystem, it is critical for organizations to create a coherent SAS workflow that can work with different analytical platforms. With different organizations using the different platforms, ability to execute diverse models with independence, traceability, and reusability becomes imperative. We will demonstrate on how SAS can be leveraged to create an intelligent workflow that can support different ecosystems and technologies. This includes interactions of Hadoop (Hortonworks, Cloudera, etc.), Teradata, SQL server and different analytical platforms to create a seamless SAS workflow which is flexible and scalable and extensible.



In today's world, every customer activity is captured on a real-time basis. The increase in data led to store data with reduced cost and in an efficient manner, and thus it introduced the Big-Data environment. The Predictive models consuming the big data should also be maintained in the same way. So, this efficient arrangement of predictive models makes us think of many questions.

- Do we need to deploy models independently?
- Does the process support the frequency of daily runs or monthly runs?

- Does the process let us know the health/rebuild of models?
- What analytical platforms (open source/Licensed) are compatible and how well all the platforms are integrated?
- Do I get a monthly summary of active models? How are the logs maintained?

INTRODUCTION

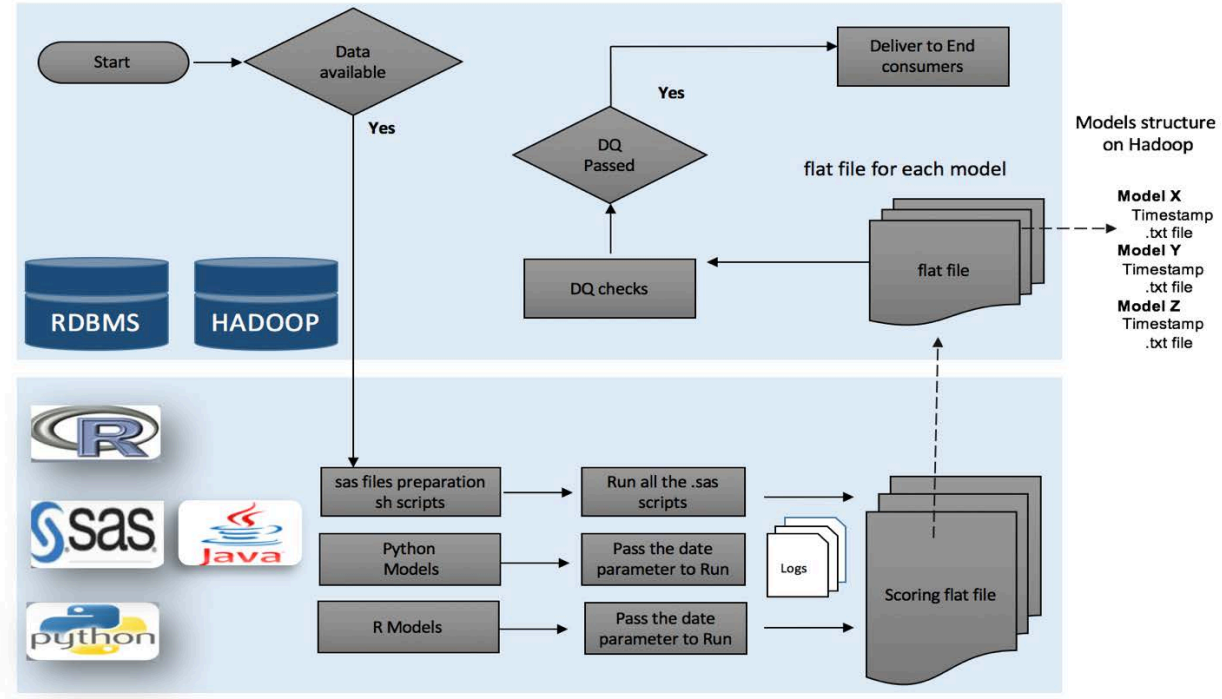
Organizations are encouraging the big data environment as it is providing the flexibility to handle and manage the structured and unstructured data. There are diverse Analytical platforms that also evolved because of the increase in voluminous and complex data. These platforms provide predictive and prescriptive solutions that address to uncover patterns, customer preferences, support strategies. Also, It is vital that we need to have a unified efficient workflow that effectively maintains and monitors the predictive models that were built by these diverse analytical platforms. Considering the evolution of various analytical platforms, We created a seamless SAS workflow which provides support to other analytical platforms

Secondly, Most organizations used to maintain a single RDBMS table that is the source of all the predictive models. This kind of layout raises the issues like interruptions, non-supportive to diverse models, flexibility, slower executions

Through this paper, we are going to elaborate on how to efficiently store and monitor predictive models independently on Big data environment using SAS intelligent workflow. Also, We will provide its support to other analytical technologies.

The high level design of scoring process:

SAS SCORING FLOW



PREDICTIVE MODELS LAYOUT AND ITS STORAGE ON BIG DATA

SAS has the capability to execute the models that are developed in other analytical platforms. This feature and SAS ease access to distributed technologies permitted us to build an intelligent workflow. The models will be deployed independently following a similar structure. And, The output will be produced in the form of flat files. These files are produced on SAS servers using analytical tools and then exported to Hadoop environment using SAS-Hadoop connectors. This efficient workflow provides the flexibility to run the individual models at their own frequency(daily, weekly).

The SAS scoring generates the flat file with the appropriate fields and will load efficiently on Hadoop environment using the Hadoop connectors(PROC HADOOP).

```
 /*  SAS generates the flat files in the below way*/
data _null_ ;
retain  << Model fields, scoring fields, ID fields>>. /* It retains the order of fields on flat file*/
                                                    ;
   set scored_data(keep=<<< Model fields, scoring fields, ID fields>>
                                                    );
   FILE  "/sas/data/&Identifier_model._&scoring_Date._&Timestamp..TXT" DLM='|' DSD;   /* Output Text File */
   PUT (_all_) (+0);
run ;

/*  Hadoop connector*/

filename configfile  "  <path>/hadoopconfig.xml"

proc hadoop cfg = configfile  ;

        hdfs mkdir = "" ;

        hdfs copyfromlocal = "/sas/data/&Identifier_model._&scoring_Date._&Timestamp..TXT

        out = <hadoop path>//&Identifier_model./&scoring_Date./&Identifier_model._&scoring_Date._&Timestamp..TXT " overwrite ;
```

BIG DATA AND MODELS LAYOUT

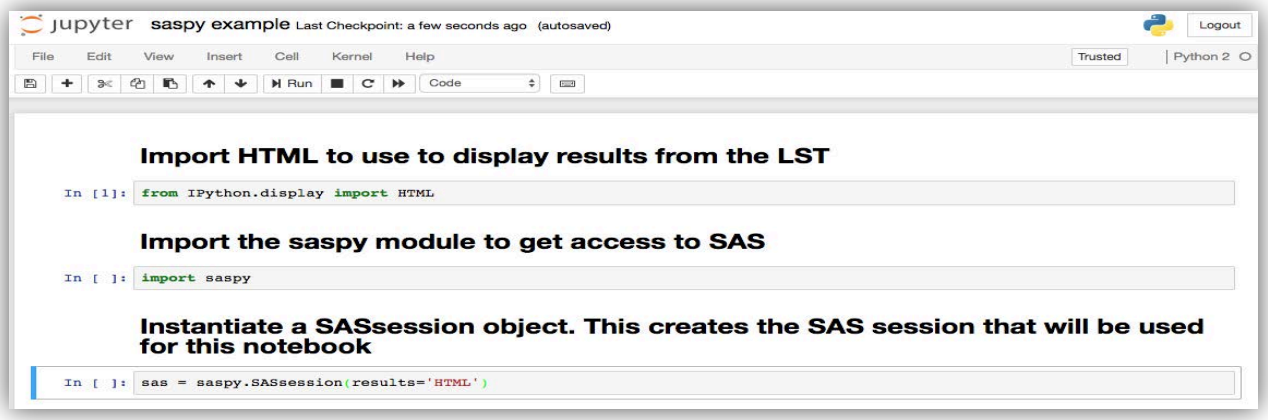
All the model flat files contain necessary identifiers and scores. These are structured consistently to avoid any confusion to the end stakeholders. These flat files at the end of scoring are pushed from SAS servers to big data systems. Each model is given a unique folder with corresponding periodic subfolder structures to enable easy identification. The model paths then can be retrieved using corresponding Hadoop commands. They can easily be loaded into a simple table which can later serve as the placeholder for all the model paths and other metadata information of the models. This structure enables scalability as well as agility with independent execution. The historic record of models is also supported and it provides an easy knob to access the scores of required time periods with ease. It has an advantage over the single horizontal structure as it allows to expose only the models required for the individual stakeholder. It is secure and fast for internal consumers who would like to perform quick analysis as well. Depending on the requirements there is a provision to custom create a single table with all the required model scores which are most frequently used by various end stakeholders.

OTHER ANALYTICAL TECHNOLOGIES

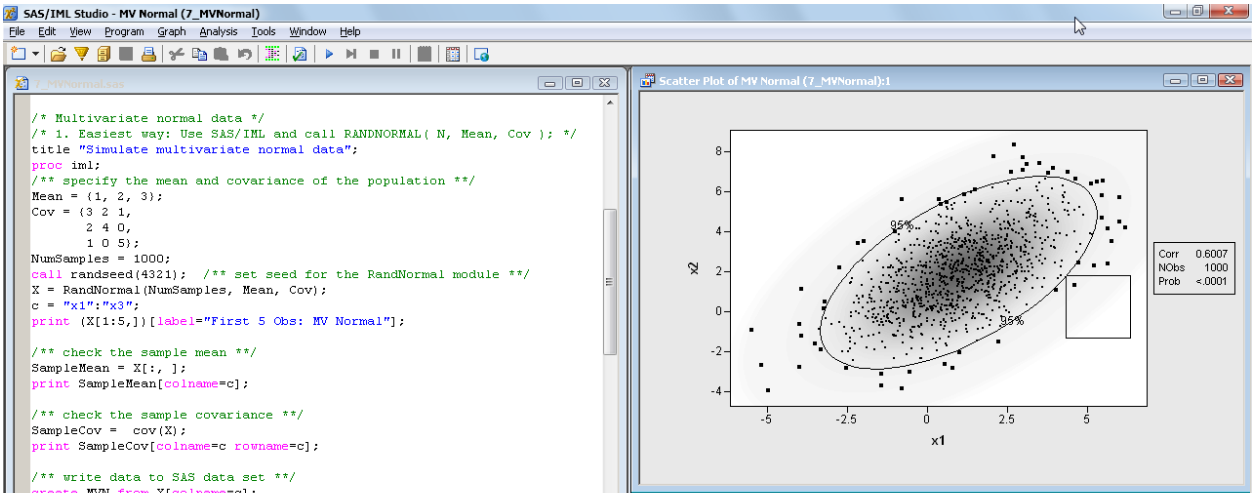
SAS is a platform that provides the flexibility to execute the programs of other languages like R and Python. This diversity provides the modeler to deploy the code in their own language. Moreover, The Hadoop connector moves the data securely, fast and reliably. This support allowed us to deploy the various models on Hadoop environment in an ease manner.

```
/*sas command to execute python*/

        data x1;
        x "python modelfile.py";
        output;
        Run;
        < SAS hadoop connector>

/*sas command to execute R*/


    submit / R;
        <Modeling code>;
        endsubmit;
        < SAS hadoop connector>
```

SAS support to Python:



SAS support to R using proc IML studio:



A Sample screenshot from SAS

SAS support to Java :

- JavaObj
- X command
- SYSTASK

SAS provides the high-performance procedures that bring in the ability to execute the modeling frameworks in a distributed way. This feature allows to run the application on multiple concurrent threads and finishes the job faster than the traditional procedures. The HP4SCORE procedure is specifically designed for scoring where it distributes and executes the jobs/single job on multiple cluster nodes.

SAS® 9.4 SAS HIGH PERFORMANCE PROCEDURES

High Performance Statistics	High Performance Data Mining	High Performance Econometrics	High Performance Forecasting	High Performance Optimization	High Performance Text Mining
HPLOGISTIC HPGENSELECT HPREG HPLMIXED HPNLMOD HPSPLIT HPFMM HPCANDISC HPPRINCOMP HPPLS HPQUANTSELECT	HPREDUCE HPNEURAL HPFOREST HP4SCORE HPDECIDE HPCLUS HPSVM HPBNET HPTSDR	HPCOUNTREG HPSEVERITY HPQLIM HPPANEL HPCOPULA HPCDM	HPFORECAST	OPTLSO Select features in OPTMILP OPTLP OPTMODEL OPTGRAPH HPCDM	HPTMINE HPTMSCORE
Common Set: HPDS2, HPDMDB, HPSAMPLE, HPSUMMARY, HPIMPUTE, HPBIN, HPCORR					

SAS high performance procedures

MODELS MONITORING

Monitoring is one of the key element in modeling lifecycle. Periodically, we need to monitor our models to factor for predictability changes, input data changes, degradation. The model performance reports play a vital role to preview the historical metrics and make pro-active measures to avoid the problems that are caused by obsolete models. Customized automation workflows can be created using deployment schema mentioned above. Discrete data sources can be connected using SAS to create the base table for performance checks. All the desired tracking metrics can be created combining the target definitions with the model scores obtained from the flat files. A similar output structure can be adopted for capturing the model metrics in flat files. Custom dashboard that automatically capture the below metrics can be built for reporting and tracking model health.

1. Input and output variable shifts distribution
2. Performance metrics like KS, PSI, lift chart, historical response rate
3. Notifications, Threshold options

SAS also provides the efficient templates for model monitoring and they are fully customizable. These templates are provided on the tools like SAS Visual Studio, SAS Model Manager.

LOG MAINTAINENCE AND MODELS FREQUENCY

The SAS logs are an excellent source to monitor the execution of all the models in the workflow. We generated a procedure that tracks the errors, warnings, run times for each of the models on a tracking table. This procedure allowed us to record performance statistics, monitor the execution of sas jobs in a transparent manner. The tracking table will maintain the history logs for each of the model. And also, It enabled us to fulfill the goal of examining and fixing the sas flows that are time-consuming, resource-intensive and fixing errors with ease.

A sample code for storing logs at a specific location and are then processed with a macro to store results on tracking table:

```
PROC PRINTTO LOG=LOG;
RUN;
```

```
proc sql;
Create table tracking_table
(
 Developername string,  -----  Modeler Name
 Identifier_model string, -----  Unique identifier for the model
 scoring_Date string,  -----  Data date for which it it scored
logfile string,  -----  Log file
status string,  -----  Execution status
error_log string,  -----  Error description
textfile_log string,  -----  Flat file issue
hadooppush_log string,  -----  Hadoop connector status
warning_log string,  -----  Warning status
Totalexecutiontime num)  -----  Execution time of a model
 quit;
```

Tracking table is maintained to get the status of execution flows and it is captured the below way :

```
data read_txt;
set read_txt;
MODELIDMATCH= substr("&var_n.",find("&var_n.",'EBI_N'),14);
if type in ('ERROR','WARNING','NOTE') then status1='ERROR-Recheck the code';
else if type in ('HADOOPPULLDONE','TXTFILECREATED')  then status1='SUCCESS';
else status1='UNKOWN/ABORTED';
if type in ('ERROR') then errorlog=Linetext;
else if type in ('WARNING') then  errorlog=Linetext;
else if  type in ('HADOOPPULLDONE') then  errorlog='NO ERROR';
else if  type in ('TXTFILECREATED') then  errorlog='NO ERROR';
else errorlog='UNKNOWN-Check it';
run;

data mapping;
set read_txt;
if status1 in('SUCCESS') then status='SUCCESS' ; else status='ERROR';
if type in ('WARNING') then warning_log=Linetext; else warning_log='';
if type in ('ERROR') then error_log=Linetext; else error_log='';
if type in ('TXTFILECREATED') then textfile_log=substr(Linetext,1,index(Linetext,' "/app')); else textfile_log='';
if type in ('HADOOPPULLDONE') then hadooppush_log=Linetext; else hadooppush_log='';
```

The independent execution feature allows the model to run at their own frequency. The hierarchy of models structure allows us to clearly identify the latest flat file of multiple iterations. So, we can run the model frequently(weekly/daily) and can obtain the latest scoring flat file of a specific duration(weekly/daily).The flat files can be easily consumed by the consumers with ease because of their independent layout on distributed systems.

SAS supports the automatic execution of jobs using the SASGSUB command. This feature provides the scope of running the jobs at different priority level, job status, optimizing grid nodes.

```
sasgsub -METASERVER <serverinfo> -METAPORT <portinfo> -METAUSER <userinfo> -METAPASS <userinfo> -GRIDAPPSERVER <serverinfo>  -GRIDWORK <workpath> -GRIDSUBMITPGM <submission path> -
GRIDJOBOPTS 'queue=normal'.     /* These options allow us to prioritize the jobs*/
```

CONCLUSION

We conclude that an efficient and unified workflow is needed as it facilitates the deployment, maintenance, and traceability. Moreover, above demonstrated kind of workflow is needed in the current organizations as it supports the diverse technologies and big-data & traditional databases. The workflow stores the model's output in the form of flat files and each model run produces a flat file. These flat files are flexible enough to be used for extraction, tables feed, market consumption. Workflow provides the flexibility to monitor and replace the models with ease. The advantage of independent monitoring and deployment allows us to deliver, rebuild the models in a quick and competent way. Through this approach, we can deploy the models at any point of time without interfering other models. Due to its independent structure there an added advantage of fine-tuning the model executions independently at different frequencies.

Unified workflow is needed as it organizes diverse analytical platforms. The increase in technologies and eco-systems are bringing in a lot of changes in the organization. These changes need to be incorporated into workflow on a constant basis to make an efficient process and support diverse platforms.

REFERENCES

https://support.sas.com/documentation/cdl/en/indbug/68170/PDF/default/indbug.pdf

http://support.sas.com/documentation/cdl/en/gridref/67371/HTML/default/viewer.htm#p0bjesvjde359nn1bfikmlzfk80b.htm

http://support.sas.com/documentation/cdl/en/imlug/66845/HTML/default/viewer.htm#imlug_r_toc.htm

https://support.sas.com/documentation/cdl/en/stathpug/68163/PDF/default/stathpug.pdf

RECOMMENDED READING

- *Base SAS® Procedures Guide*
- *Hadoop for SAS connectors and information*
- *SAS Visual Studio*
- *R ,Python , Java Integration on SAS*
- *Hadoop environment*
- *SAS High performance procedures*

CONTACT INFORMATION

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

Anvesh Reddy Minukuri
Comcast Corporation
(405) 780-5346
anveshreddy_minukuri@comcast.com

Anvesh Reddy Minukuri is a Lead data scientist at Comcast. Before joining at Comcast, He worked for Mahindra Satyam (Indian MNC) as a Software Engineer. He pursued his masters in OSU Data Mining Master's program from Oklahoma State University and Computer engineering from JNTU University, India. He joined Comcast in 2015 and is supporting EBI in diverse Analytical projects. His areas include marketing models, 360 customer view, customer Persona and retention strategies. He has published papers for SAS global forum and Analytical conference.

Ramcharan Kakarla
Comcast Corporation
(267) 283-7395
ramcharan_kakarla@comcast.com

Ram Kakarla is currently Lead Data Scientist at Comcast. He holds a master degree from Oklahoma State University with specialization in data mining. Prior to OSU, he received his bachelors in Electrical and Electronics Engineering from Sastra University.

In his current role he is focused on building predictive and prescriptive modeling solutions around marketing challenges. He has several papers and posters in the field of predictive analytics. He served as SAS Global Ambassador for the year 2015

Sundar Krishnan
Comcast Corporation
(404) 754-7112
sundar_krishnan@comcast.com

Sundar Krishnan is currently Lead Data Scientist at Comcast. He is passionate about Artificial Intelligence and Data Science. He completed his masters from Oklahoma State University in Management Information

System with a specialization in Data Mining and Analytics. He focuses on 360° customer analytics models at Comcast. His project experience includes marketing based models, machine learning workflow automation, image classification and language translator.