Paper 823-2017

A Hadoop Journey Alongside the SAS®-paved Road

Dmitriy Khots, Jeremy Wortz, Sumit Sukhwani, Krutharth Peravalli, Amit Gautam

West Corporation

ABSTRACT

As the open-source community has been taking the technology world by storm, especially in the big data space, large corporations such as SAS, IBM, and Oracle have been working to embrace this new, quickly evolving ecosystem to continue to foster innovation and to remain competitive. For example, SAS, IBM, and others have aligned with the Open Data Platform initiative and are continuing to build out Apache Hadoop and Apache Spark solutions. And, Oracle has partnered with Cloudera to create the Big Data Appliance. This movement challenges companies that are consuming these products to select the right products and support partners. The hybrid approach—using a combination of tools available—seems to be the methodology chosen by most successful companies. West Corporation, an Omaha-based provider of technology-enabled communication solutions, is no exception. West has been working with SAS for over 10 years in the ETL, BI, and advanced analytics space, and West began its Hadoop journey a year ago. This paper focuses on how West data teams use both technologies to improve customer experience in the interactive voice response (IVR) system by storing massive semi-structured call logs in HDFS, by experimenting with building models that predict a caller's intent to route the caller more efficiently and to reduce customer effort using familiar SAS code and the user friendly SAS® Enterprise Miner™, and using PySpark for large scale modeling efforts that adapt the code initially developed in SAS.

INTRODUCTION

West Corporation provides automated communication solutions and enables connected customer experiences for some of the world's most recognizable brands. When consumers and businesses interact with West platforms and applications, a great variety of data is generated, including information about who interacted, when the interaction happened, why it happened, and what happened during the interaction. This data is then used to provide operational reporting and insights as well as drive improvements in customer experience and optimize performance of the applications.

West generates roughly 1TB of such business data per day across hundreds of various applications and data sources. Historically, this data has been managed using traditional RDBMS-based data warehousing technology and consumed by various BI and analytics tools such as SAS through ODBC/JDBC protocols. However, it's become increasingly complex, both from a technical and an economic perspectives, to continue to manage the three V's (volume, velocity, variety) of Big Data using traditional means. Also, it has become computationally costly to process massive amounts of data while it traverses through networks that connect geographically dispersed data centers. As one systems engineer puts it, "physics matters!"

One word that has become synonymous with the term Big Data is Hadoop. While Hadoop is no longer any one single concept, and rather a collection of open source and proprietary projects and software packages, at its core, it is based on an open source, fault tolerant (self-healing), scalable, secure, distributed file system equipped with computational frameworks for fast ingest (schema-on-read vs schema-on-write) and processing (MapReduce and Apache Spark) of Big Data and has become the industry best practice for Big Data management and analytics. Co-location of storage and compute (aka distributed computing) as well as the economics side of the equation are further drivers for adoption of Hadoop in the industry and at West, who has partnered with a leading enterprise-grade Hadoop distributor to start the Hadoop journey.

While Hadoop has recently celebrated 10 years, it is still fairly new and while the Apache Software Foundation community has thousands of contributors (most brilliant minds across the world) who accelerate the speed of enterprise adoption, this technology has some catching up to do in terms of end user experience and ease of coding compared to some of the proprietary tools in the market like SAS. The Center for Data Science at West have been loyal SAS customers for the past 11+ years, with a variety of certifications and training in tools such as Base SAS, Advanced SAS Programming, SAS Macro Facility, SAS Enterprise Guide and SAS Enterprise Miner and hence the team is highly efficient in delivering quality code for predictive modeling and similar analytical applications.

This paper explores West journey at operating with Big Data using a combination of SAS and Hadoop. SAS is used due to coding preferences and the ability to manipulate sample data quickly, while Hadoop is used for execution speed across a large clustered environment. The use case for this study is a predictive modeling exercise with IVR data, including building models that predict a caller's intent which are used to optimize call routing inside the automated phone system and improve customer experience.

In particular, this paper provide an overview of West IVR system and related data (IVR logs), West SAS and Hadoop deployments, a review of IVR log pipelines into SAS and Hadoop, a review of data preparation and target identification processing occurring in SAS, modeling efforts in SAS Enterprise Miner to test out various modeling parameters and variable selection algorithms, and reviewing the PySpark (Apache Spark Python API) and Spark modeling efforts that transitioned the initial SAS development into Hadoop production framework.

INTERACTIVE VOICE RESPONSE SYSTEMS

An Interactive Voice Response System is the front line to most customer interactions with a brand. The purpose of the system is to automate as many routine tasks as possible (e.g. paying a bill or placing an order), gather valuable data about the caller and transfer the caller to a live representative along with that data (in order to avoid repeated questions and reduce customer effort), if the IVR is not able to serve the customer. Typical customer experience in IVR is shown in Figure 1.

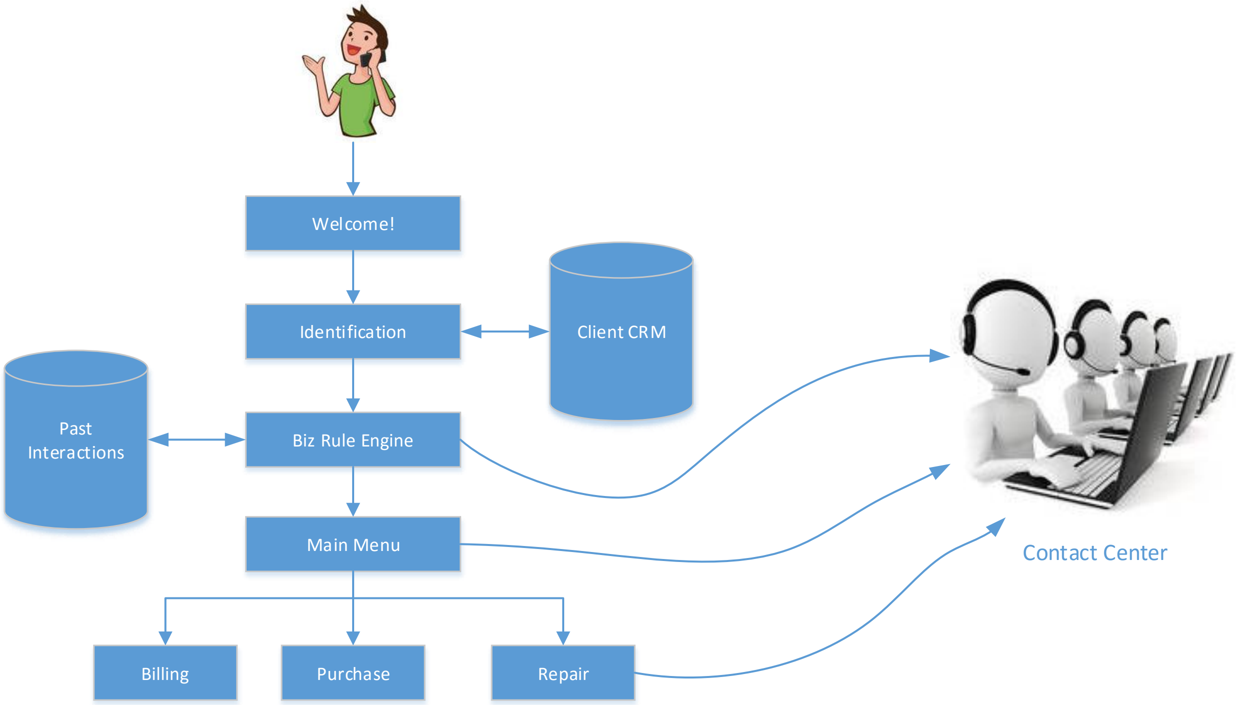

Figure 1. Simplified IVR Call Flow

Customers who are "repeat callers" expect the system to know who they are and why they are calling the next time they call in, so that they can have the preferred experience in the system and/or with a live agent. West uses various means to ascertain a caller's intent without having to ask the purpose of the call. This includes using past interactions and a business rules engine to create business rules that drive predictive intent questions, e.g., "are you calling about your bill?" If the customer says "yes", then the customer does not have to navigate throughout the IVR options and gets directly where he or she would like to go; however, if the customer says “no”, then customer effort actually increases and experience is suboptimal.

Accuracy of predictive intent logic is thus a critical aspect of a successful deployment. Simple business rules may not provide desired levels of accuracy, however, predictive models can. Figure 2 shows the value of a highly accurate predictive intent model that can leapfrog the customer into the right place in IVR and reduce customer effort.

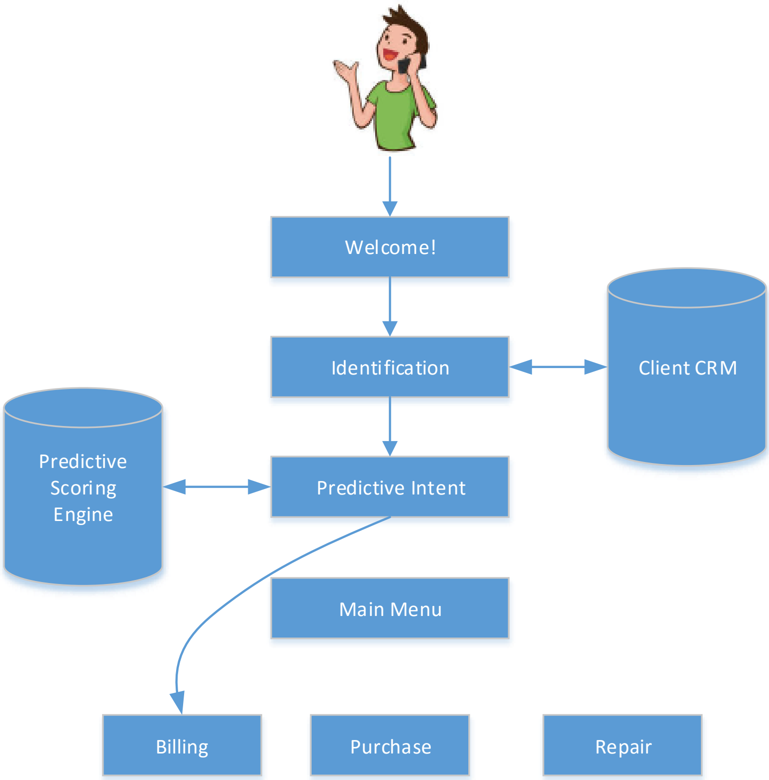

Figure 2. Simplified IVR Call Flow with Predictive Intent

The critical element of building accurate predictive intent models is the data generated by IVR. West transactional IVR logging platform captures all events that occur in the system while the customer is traversing through a call path, including messages played by the system, customer responses, auxiliary API hits, etc. For more details on IVR system and data, see Khots 2015. The API return data is very rich and semi-structured – this is the data that is the prime candidate for storage and analytics on Hadoop.

INFRASTRUCTURE OVERVIEW

This section provides a high level overview of West SAS and Hadoop footprints.

SAS ECOSYSTEM

West has a consolidated SAS footprint which consists of the application server running all main SAS components and services and a load-balanced remote desktop server that can handle up to 20 simultaneous users. End users connect into RDS through an RDP connection, ensuring all information is accessed in a secure fashion. SAS administrators establish data connections by creating DSNs on the application server, this allows end users to query data into SAS from source systems using SAS/ACCESS to ODBC and PROC SQL pass-through facility. Figure 3 shows the high level data flow as well as various SAS components that are utilized by end users. Primary SAS components running on the application server are Base SAS (and related components), SAS/OR, SAS/ACCESS to ODBC, SAS Studio Server, SAS Enterprise Miner, and SAS Enterprise Guide. RDS runs just the related thin clients and Web UI's.

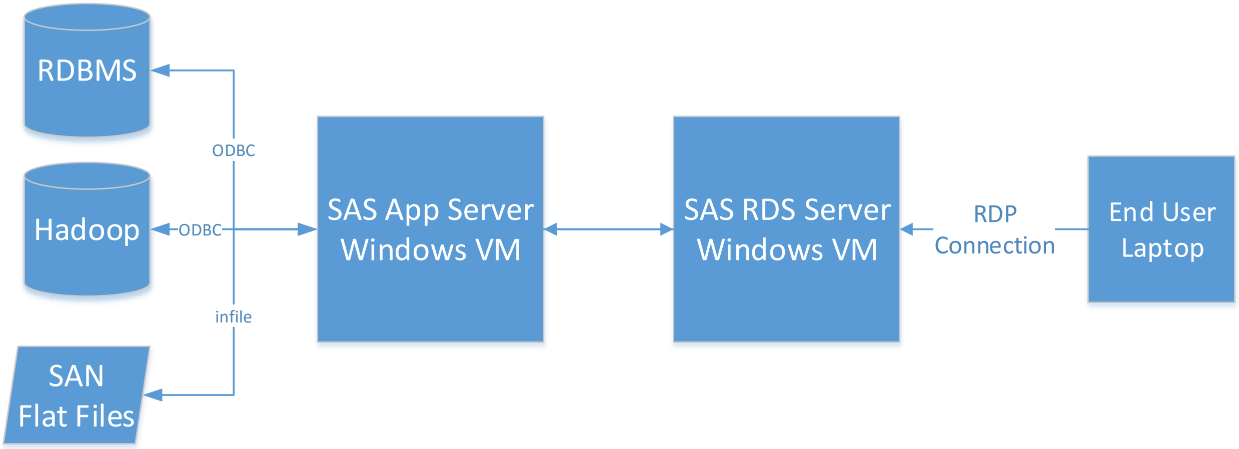

Figure 3. West SAS Footprint

Traditional pattern for analytics and predictive modeling using SAS has been extracting all data from the source systems into the SAS environment and working with SAS data sets. This works well for smaller data, however, when source system resides in a different data center and is fairly large, then pulling across the network is not efficient. Another solution is needed. This is what is offered by Hadoop and SAS Data Loader and Scoring Accelerators for Hadoop. Review of these solutions is outside the scope of this paper and will be tackled in future efforts. This paper offers a workaround for end users who are familiar with SAS code (for fast and agile development), yet want to run the final results on Hadoop due to data sizes.

HADOOP ECOSYSTEM

West Hadoop journey began with a successful “Proof of Concept” cluster powered by the best-in-class Hadoop distribution. Batch MapReduce and Spark processing, interactive SQL capabilities, fast ingestion of new data sources, fast data consumption through existing BI technology via ODBC, and advanced machine learning with Apache Spark ML, all performed on billions of observations in a lightning fast manner, running just on a smallish ten node cluster have all been used to create a business case for production deployment.

West Hadoop production deployment now consists of a much larger physical fourteen node cluster that has considerable compute, memory, and storage capabilities alongside a smaller virtual twelve node cluster for pre-production purposes. This is just the beginning. Additional use cases will be driving substantial expansion of the infrastructure over the next 12-18 months. Figure 4 shows the high level components on both production and pre-production Hadoop clusters.

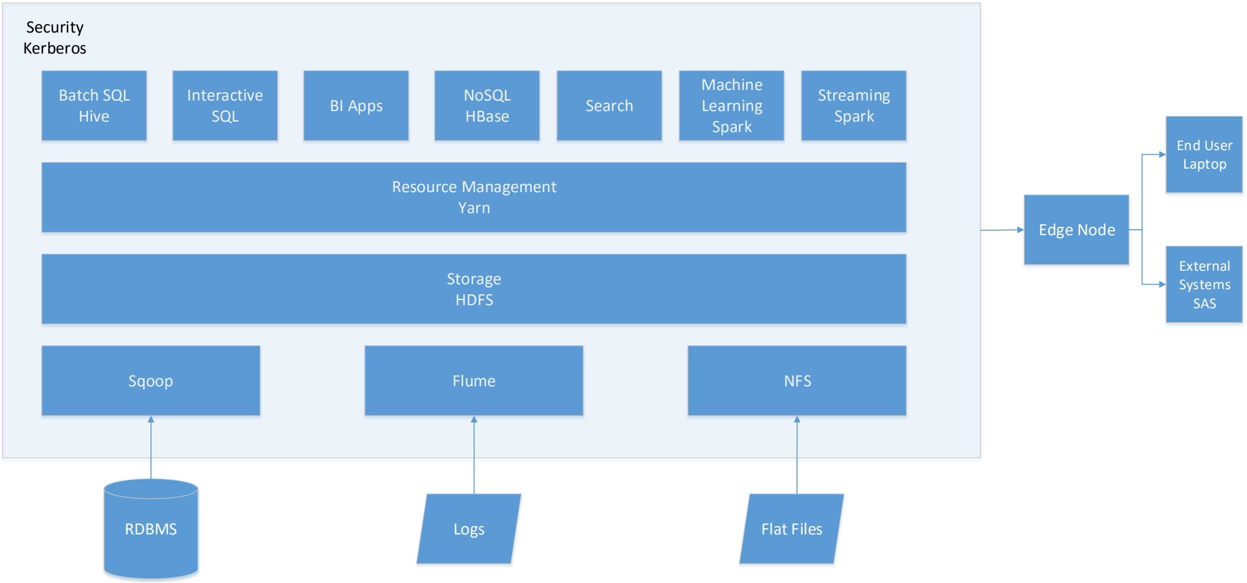

Figure 4. West Hadoop Footprint

ANALYTICS FRAMEWORK

Figure 5 provides an overview of the data mining framework for predictive intent in IVR utilizing both SAS and Hadoop.

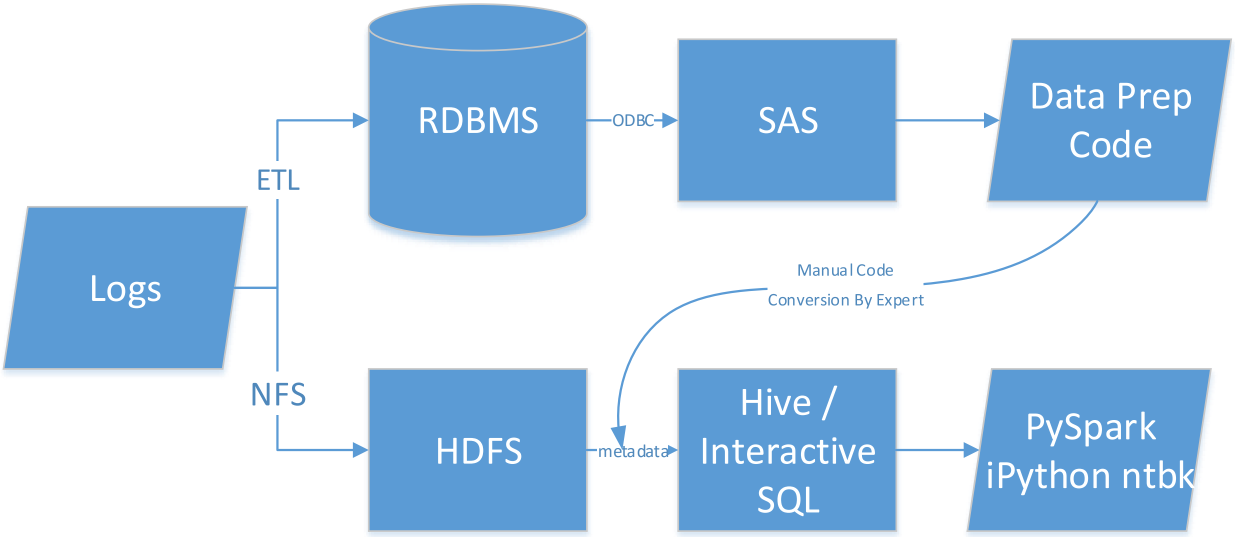

Figure 5. Predictive Intent Framework

The upper part of Figure 5 is what typically occurs for analytics purposes. ETL technology is used to ingest IVR logs into a relational database in an MPP engine and data is then moved once again from RDBMS into SAS via SAS/ACCESS to ODBC. For the purposes of this study, source IVR logs are concurrently loaded into the legacy RDBMS MPP engine as well as into HDFS via a cron job that continuously executes *hdfs dfs put* statements from an NFS mount.

DATA PREP AND ANALYSIS IN SAS

MPP engine stores data in third normal form. The SAS code below is used to retrieve data via ODBC into a SAS data set. The resulting dataset is a denormalized list of all events that occurred on all IVR calls for a specific set of callers.

```
/* Pass-through PROC SQL Facility to ingest data from MPP into SAS */
Options compress=yes;

PROC SQL;
        CONNECT TO ODBC as myODBC (NOPROMPT= "DSN=MPP");
        CREATE TABLE Data_input AS
        SELECT * FROM CONNECTION TO myODBC
        (SELECT &field_selection
        from txl.Transaction_call_fact ivr
        left join txl.transaction_pnc_fact pnc on (pnc.wic_ivr_key_identifier =
        ivr.wic_ivr_key_identifier)
        left join txl.client_customer_attribute ca on (ca.wic_ivr_key_identifier
        =ivr.wic_ivr_key_identifier
         and ca.customer_attribute_name='CAT_FinalAcct')
        left join txl.nodename_dim node on (pnc.nodename_identifier = node.nodename_identifier)
        left join ods.program_apn apn on (ivr.program_apn_identifier =
        apn.program_apn_identifier)
        left join ods.ods_client cli on (apn.client_identifier = cli.client_identifier)
        left join ods.program prg on (prg.program_identifier = apn.program_identifier)
        left join txl.transaction_prompt_status tps on
        (pnc.transaction_prompt_status_identifier=tps.transaction_prompt_status_identifier)
        left join txl.response_type rt on
        (pnc.response_type_identifier=rt.response_type_identifier)
        left join txl.call_exit_type cet on (ivr.call_exit_identifier = cet.call_exit_identifier)
        left join txl.transaction_transfer_fact ttf on (ivr.wic_ivr_key_identifier =
        ttf.wic_ivr_key_identifier)
        left join txl.transfer_reason tr on
        ttf.transfer_reason_identifier=tr.transfer_reason_identifier)
        where cli.client_number = 1111 and prg.program_number = 2222  ivr.start_date >=&Date1 and
        ivr.start_date < &Date2 and extract(hour from ivr.start_time)= 5);
QUIT;
```

Then a sequence of transformations is applied using PROC SQL and DATA STEP to create a set of predictor variables, including N previous call reasons (using the RETAIN statement in a DATA STEP) and the categorical target variable (reason for the current call, extracted from caller responses). The final deliverable is a data set that can be ingested by SAS Enterprise Miner for predictive modeling purposes. SAS Enterprise Miner is then used to test various modeling approaches and feature selection algorithms on a smaller data set to determine feasible models.

The challenge with this approach is two-fold. First, the source data is so large that it is too time consuming to extract it all from the MPP engine into SAS via ODBC. Creating a random sample within RDBMS is also resource intensive. Second, once extracted, it is physically impossible for SAS Enterprise Miner running on the current SAS application server, given its computational resource constraints, to completely run all the predictive model iterations that need to be run. More horse power is needed and, hence, running this code on Hadoop (after it is manually recreated in Python and/or Spark) is a feasible option.

DATA PREP AND ANALYSIS IN HADOOP AND SPARK

Data management for predictive modeling purposes on Hadoop is a four stage process:

1) Raw IVR logs are landed into HDFS and partitioned by date using MapReduce.
2) HDFS data is converted into Parquet format.
3) HDFS data is converted into denormalized SQL metadata structure on HDFS.
4) Logic derived using SAS analysis is used to create new SQL metadata structures on HDFS (this creates predictor and target variables).

Final predictive modeling step is accomplished using PySpark and/or Spark, which are used for feature selection and predictive modeling on the large datasets in HDFS.

An example of SQL metadata coding on Hadoop used in step 4 is below:

```
create table transform.predictive_prep_step1 as
select select_field_list,yyyymm
from denorm.detailed_tier_analysis
where start_date>=Date1 and start_date<= Date2;
```

For this study, the Jupyter notebook has been used to develop PySpark code. The PySpark ML API was used due to its beneficial features such as dataframe support and the concept of pipelines for quick machine learning development (as opposed to the older MLLIB API). Before ML Pipelines can be built, pyspark.ml package components are invoked using the following code:

```
from pyspark.ml import Pipeline
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.feature import VectorIndexer
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.mllib.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.feature import OneHotEncoder, StringIndexer
from pyspark.ml.classification import LogisticRegression
from pyspark.mllib.evaluation import MulticlassMetrics
```

Two more examples of PySpark are provided below. First example is defining a random forest classifier object that sits within the pipeline. Prior steps in the pipeline include data prep and manipulation steps, commonly found in most predictive modeling exercises.

```
rf = RandomForestClassifier(labelCol="label", featuresCol="featuresIndexed", numTrees=60,
                            featureSubsetStrategy = "onethird",
                            maxDepth = 7, maxBins = 32, impurity = "entropy")
```

Here is an example of a logistic regression pipeline step.

```
lr = LogisticRegression(maxIter=200, regParam=0.0001, elasticNetParam=0.1)
```

One key benefit of this approach, from a computational perspective, is that data is physically moved only once, once it lands into HDFS from NFS. Everything else occurs in metadata and in memory. This saves considerable computation time, but does require additional coding / developer time. This process worked for West at the onset of Hadoop journey due to the fact that data scientists who were used to SAS had more expertise with the specific IVR log data compared to data scientists who were used to Hadoop. It was a fruitful collaboration, where SAS users would quickly perform data preparation in SAS, while Hadoop users were concerned with developing PySpark code, rather than taking time to create new logic for predictor variables.

Another key benefit for this approach is the ability for hyper-parameter tuning in the pipeline. Meaning: once a pipeline was created, defining an entire predictive modeling process, the parameters of the models could be tested to find the ideal model fit. Specifically, for the random forest example, a grid of 50, 100, 150 trees, along with impurity measurements of gini, entropy, created a grid of 3 x 2 options = 6 combinations, where the ‘winner’ model was selected. This provides greater efficiency for the modeler to create an automated methodology where the best model over a complex combination of hyper parameters is selected. This out-of-the box functionality provides an easier way to find the optimal model as opposed to SAS EM (which can be done, but is sometimes cumbersome). See example code below:

```
lr = LogisticRegression()
grid = ParamGridBuilder().addGrid(lr.regParam, [0.01, 0.001, 0.0001]).build()
evaluator = BinaryClassificationEvaluator()
cv = CrossValidator(estimator=lr, estimatorParamMaps=grid, evaluator=evaluator)
cvModel = cv.fit(trainingData)
```

The PySpark models were executed in 30 minutes (running on hundreds of millions of rows and hundreds of predictor variables) and generated a KS of 30+, whereas SAS Enterprise Miner modeling process did not finish after a few days. This was in large part due to compute resource differences between Hadoop and SAS environments (number of cores). The Hadoop cluster had 10x more cores allocated to worker nodes relative to the SAS environment. This also leads into the economic benefits of using Hadoop for large computations.

CONCLUSION

West Center for Data Science has been able to adopt new data management and analytics technology and continue to utilize existing methods and tools to expedite completion of work. After all, this is the key in business – solving problems using available resources in a timely and cost effective manner. West is going to continue to explore open source technology for Big Data to drive efficiencies, discover and deploy new use cases and continue to work with SAS in order to facilitate advancement of analytics and data driven insights for West and its customers.

While Hadoop and related technology is very promising, its adoption rate across the enterprise, for various production purposes remains to be seen. West and CDS are excited about this journey and to discover where it will lead us. Future research will include piloting SAS solutions that are integrating the technology with open source, including SAS Data Loader for Hadoop, SAS Scoring Accelerator for Hadoop and SAS Viya.

REFERENCES

Khots, Dmitriy. 2015. Unstructured Data Mining to Improve Customer Experience in Interactive Voice Response Systems. *Proceedings of the SAS Global Forum 2015*, Dallas, TX, SAS. Available at https://support.sas.com/resources/papers/proceedings15/3141-2015.pdf.

ACKNOWLEDGMENTS

The authors would like to thank Topher Wren for his vision, leadership and guidance in all strategic efforts for the team, we would also like to thank Earl Hooper, Dilip Sarangdevot, Dan Linder, Bryan Melvin, Priya Srinivasan and Shruti Ramesh for their contributions to architecture and deployment of Big Data infrastructure, data engineering, and data science guidance. The authors would also like to thank Dr. Goutam Chakraborty for inviting the team to present this topic at SAS Global Forum 2017.

CONTACT INFORMATION

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

Dmitriy Khots, Ph.D.
West Corporation
+1.402.716.0766
dkhots@west.com
www.west.com