Efficient Deployment of Predictive Analytics through Open Standards and Cloud Computing

Alex Guazzelli
Zementis, Inc.*
Alex.Guazzelli@zementis.com

Kostantinos Stathatos
Zementis, Inc.
Kostas.Stathatos@zementis.com

Michael Zeller
Zementis, Inc.
Michael.Zeller@zementis.com

ABSTRACT

Over the past decade, we have seen tremendous interest in the application of data mining and statistical algorithms, first in research and science and, more recently, across various industries. This has translated into the development of a myriad of solutions by the data mining community that today impact scientific and business applications alike. However, even in this scenario, interoperability and open standards still lack broader adoption among data miners and modelers.

In this article we highlight the use of the Predictive Model Markup Language (PMML) standard, which allows for models to be easily exchanged between analytic applications. With a focus on interoperability and PMML, we also discuss here emerging trends in cloud computing and Software as a Service, which have already started to play a critical role in promoting a more effective implementation and widespread application of predictive models.

As an illustration of how the benefits of open standards and cloud computing can be combined, we describe a predictive analytics scoring engine platform that leverages these elements to deliver an efficient deployment process for statistical models.

1. INTRODUCTION

The data mining community has derived a broad foundation of statistical algorithms and software solutions that allowed for statistical analysis to become a standard approach used in science and industry [16].

Much emphasis has been placed on the development of predictive models, which usually encompasses a lengthy data mining and analysis phase followed by model development and evaluation. As a consequence, currently, the market place offers a range of powerful tools, many open-source, for effective model building.

However, once we turn to the deployment and practical application of predictive models within an existing IT infrastructure, we face a much more limited choice of options. Often it takes months for models to be integrated and deployed via custom code or proprietary processes.

Effective integration and real-time execution as part of a broader Enterprise Decision Management strategy [17] is where we will see the true value of predictive models arise.

We are today in the unique situation that various open standards and Internet-based technologies have reached maturity and are available at our disposal to provide a more effective end-to-end solution for the deployment of predictive analytics models.

Service Oriented Architecture (SOA) has become the widely accepted practice for the design of loosely coupled IT systems, e.g., implemented on the basis of Web Services [11][15]. Combined with the emerging Cloud Computing paradigm [7], the Software-as-a-Service (SaaS) license model now provides the opportunity for vendors to deliver software solutions as a cost-effective service that scales with the user's demand and is paid for based on actual consumption like your utility bill.

Most importantly for the data mining community, the Predictive Model Markup Language (PMML) standard [10] has reached a significant stage of maturity and obtained broad industry support, which allows users to exchange predictive models among various software tools.

Open standards and cloud computing not only have the power to enable the development of many more data mining applications across science and industry, but more importantly they also lower the total cost of ownership by avoiding proprietary issues and incompatibilities among systems.

To illustrate how a deployment platform for predictive analytics can be hosted in clouds, we use the ADAPA scoring engine, which leverages not only cloud computing but also open standards to offer instantaneous deployment capabilities for a variety of data manipulation routines and predictive models.

2. INTEROPERABILITY AND OPEN STANDARDS

Although deployment of data mining and statistical algorithms is an integral part of the life cycle of a data mining project [8], many of the data mining tools available today do not address the deployment of the solutions they help to build. In fact, until recently, the deployment task was viewed as the responsibility of IT. This is understandable given all the technological hurdles that needed to be overcome for successful deployment, which included the provisioning of hardware and software as well as the interchange of information between different platforms.

The advent of data mining specific open standards such as PMML as well as the emergence of cloud computing has turned this view upside-down: the deployment of models

*Zementis, Inc. (`www.zementis.com`) is located at 6125 Cornerstone Court East, Suite 250, San Diego, CA.

can now be achieved by the same team who builds them. Below, we show how these new technological elements can be pieced together to offer instantaneous, reliable, and scalable deployment of data mining solutions.

2.1 Representing Models in PMML

Developed by the Data Mining Group (`www.dmg.org`), an independent, vendor led committee, PMML provides an open standard for representing data mining models [10]. In this way, models can easily be shared between different applications avoiding proprietary issues and incompatibilities.

PMML is an XML-based language that follows a very intuitive structure to describe data pre- and post-processing as well as predictive algorithms. Not only does PMML represent a wide range of statistical techniques, but it can also be used to represent input data as well as the data transformations necessary to transform raw data into meaningful features.

2.1.1 PMML Structure

PMML specifies a well-defined list of elements that are used to describe data and models. To explain its versatility, this section will explain the main structural features of the standard. For example, all the inputs to the model are described on the "DataDictionary" element. It is in this element that an input field is defined as continuous, categorical, or ordinal. The data dictionary is also used to describe the list of valid, invalid, and missing values.

When it is time to describe the usage type for an input field, this is established in the "MiningSchema" element. Values for usage type are: active, supplementary, and predicted. It is also in this element that the appropriate treatment for missing and outlier values is specified, which is of particular importance for the production deployment of models where they may be subjected to incomplete, corrupted, or unforeseen data submitted by other applications.

Figure 1 shows a fragment of PMML code for a neural network back-propagation model trained on the well known El Nino dataset [4]. The representation of the neural network model which is called "ElNino_NN" begins with the mining schema.

```
<NeuralNetwork modelName="ElNino_NN" functionName="regression"
    activationFunction="tanh">
    <MiningSchema>
        <MiningField name="latitude" usageType="active"/>
        <MiningField name="longitude" usageType="active"/>
        <MiningField name="zon_winds" usageType="active"
            invalidValueTreatment="asMissing"
            missingValueReplacement="0"/>
        ...
        <MiningField name="airtemp" usageType="predicted" />
    </MiningSchema>
    ...
```

Figure 1: PMML mining schema exert for the El Nino neural network model. The model requires three inputs (marked "active") and returns a result as defined in the predicted field.

Note that while variable "airtemp" represents the predicted field, all other fields are required inputs, since they are marked as "active". We are also specifying in the mining schema how invalid and missing values should be treated for the field "zon_winds". In this case, invalid and missing values are being replaced by the value "0".

PMML defines a range of data transformations elements that allow for the normalization, discretization, aggregation, and mapping of values. It also offers a set of built-in functions for complex arithmetic operations and string manipulations. To achieve extensibility, PMML includes an element for user-defined functions that can be used to implement data manipulation in PMML using boolean operators. These can be further combined under an IF-THEN-ELSE structure. This combination allows for the expression of complex logic and derivation of powerful feature detectors from raw input data. [1]

Model specifics such as parameters and architecture are defined under different model elements, which cover a host of predictive algorithms, including:

- Neural Networks
- Support Vector Machines
- Regression Models
- Decision Trees
- Association Rules
- Clustering
- Sequences
- Naïve Bayes
- Text Models
- Rules

The next version of PMML, version 4.0, to be released in 2009 expands the list of algorithms supported. For example, it will include elements for time series and model segmentation. It will also extend the functionality covered by existing elements as well as provide for many ways for a model to be explained, including lift/gains charts and ROC graphs. PMML 4.0 also allows for boolean data types and expands its range of built-in functions.

2.1.2 PMML On-The-Go

The leading model development environments currently available on the market export models in PMML. Some of these tools also provide import functionality so that one is able to visualize and further refine a model that was originally produced in a different environment. Open-source environments worth mentioning here are KNIME (`www.knime.org`), which imports and exports many PMML models as well as the R project for statistical computing (`www.R-project.org`). Zementis recently contributed code to the R PMML package [18] to allow for the export of neural networks and support vector machines. Besides these two classes of models, the package also exports PMML for various other algorithms including decision trees and regression models.

[1] For a comprehensive PMML Data Pre-Processing Primer which explains in detail how data can be manipulated in PMML, please refer to our predictive analytics resources website (`www.predictive-analytics.info`).

Once a model is exported into PMML, it can easily be moved to another PMML compliant application. [2]

2.2 Executing Models via Web Service Calls

Web Services offer a simple, interoperable, messaging framework. Providing the foundation for Service Oriented Architecture (SOA), Web Services are pervasive throughout applications, operating systems, and are in effect the essence of interoperability. They use XML to code and decode data and the SOAP (Simple Object Access Protocol) standard to transport it [19]. With Web Services, data can be easily exchanged between different applications and platforms.

For the ADAPA scoring engine, the Web Services component executes models expressed in PMML through a web-service call that follows the Java Data Mining (JDM) standard [12]. JDM is a standard Java API for developing data mining applications and tools. The JDM 1.0 standard was developed under the Java Community Process as JSR 73. The ADAPA Web Service is described by a WSDL (Web Services Description Language) file defined by JDM. To connect to the Web Service, one can read the WSDL to determine what operations are available for execution and use SOAP to call one of these operations.

Figure 2 shows a SOAP request for the PMML file "ElNino_NN". In this example, we use this neural network model to score a single data record. This is accomplished through the generic "executeTaskElement" web service operation, which is defined in the WSDL file. The actual task to be executed is provided as a parameter. In this case, a "RecordApplyTask" element is provided, carrying information about the model to be used for scoring as well as the data record containing all the necessary input fields.

```
<soap:Body
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <jdmws:executeTaskElement
    xmlns:jdmws="http://www.jsr-73.org/2004/webservices"
    xmlns:jdm="http://www.jsr-73.org/2004/JDMSchema">
    <task xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      modelName="ElNino_NN" xsi:type="jdm:RecordApplyTask">
      <jdm:recordValue name="latitude">
        <jdm:value decimal="8.96" xsi:type="jdm:DecimalValue"/>
      </jdm:recordValue>
      <jdm:recordValue name="longitude">
        <jdm:value decimal="-140.32" xsi:type="jdm:DecimalValue"/>
      </jdm:recordValue>
      <jdm:recordValue name="zon_winds">
        <jdm:value decimal="-6.3" xsi:type="jdm:DecimalValue"/>
      </jdm:recordValue>
      ...
      <jdm:applySettingsName>ElNino_NN</jdm:applySettingsName>
    </task>
  </jdmws:executeTaskElement>
</soap:Body>
```

Figure 2: A SOAP request using JDM operations for the scoring of a single data record through the "ElNino_NN" model.

The response returned for such a request includes all the input fields provided plus a value for the predicted field "airtemp".

[2]Since not all tools export the latest version of PMML, version 3.2, we have made available to the data mining community a PMML Converter tool that reads in models expressed in older versions of PMML and converts them to version 3.2. The PMML Converter is available free of charge through the Zementis website (`www.zementis.com`).

2.3 Cloud Computing

The term cloud computing has recently drawn extensive attention, especially in the nontechnical business media [7], due to its promise to reduce cost and management overhead for IT. In fact, cloud computing represents yet another shift in the geography of computation. From the service bureaus and time-sharing systems of almost 50 years ago to the advent of the personal computer and shrink-wrap software in the 1980s, and finally to the Internet as a platform, we are once again changing the face of computing [14].

At its core, cloud computing is a set of services that provide computing resources via the Internet. Large data centers deliver scalable, on-demand, often virtualized, resources as a service, eliminating the need for investments in specific hardware, software, or on your own data center infrastructure. The term "cloud" is used as a metaphor for the Internet.

Cloud computing allows for a variety of services, including storage capacity, processing power, and business applications. Accessing services on the cloud is not a new concept, but it was only recently that it became available as a secure and reliable infrastructure. Today, big companies such as Microsoft, Sun, IBM, Google, and Amazon are offering storage and virtual servers that can be accessed via the Internet on demand. For an in-depth comparison of representative cloud platforms, please refer to [3] and [6].

One example of a generic cloud infrastructure is the Amazon Elastic Compute Cloud (Amazon EC2). Powered by Amazon Web Services (AWS), Amazon EC2 provides dynamic compute capacity in the cloud [1]. It is remarkable not only because of its extensive cloud infrastructure and active user community resources, but also because it provides a well defined Service Level Agreement (SLA) for availability.

The open-source community has also introduced its own infrastructure for supporting cloud computing. Examples are Sector/Sphere [13] and Hadoop [5]. The creation of the OCC - Open Cloud Consortium (`www.opencloudconsortium.org`) represents a great step forward in terms of developing standards for cloud computing and frameworks for interoperating between clouds. Inside the OCC, different working groups address not only standards for clouds, but also information sharing and security issues as well as the operation of a wide area cloud testbed for cloud computing, which currently includes various institutions throughout the United States.

SaaS (Software as a Service) is a software license model in which a business or a user may access software via the Internet and pay for the right to use the software for a certain time period rather than acquiring a perpetual software license to be installed in-house. This is extremely advantageous for customers, since there are no upfront costs in setting up servers or licensing software and it minimizes the risk of purchasing costly software that may not provide adequate return of investment. Well known commercial SaaS applications include Salesforce.com and Google Apps.

Since the SaaS license model and cloud computing are both Internet-centric, we see more and more vendors combining them to deliver novel software solutions. Below, we examine how cloud computing provides a natural infrastructure for deploying analytics using open standards.

3. PUTTING MODELS TO WORK

Given all the technologies available today that can be used

for effective model deployment, it is interesting to note that only a few systems offer the flexibility and interoperability requisites necessary for accomplishing such a task. On the model development side, Biocep (`www.biocep.net`) combines the capabilities of the R project with web-services and cloud computing aiming to provide users a web-based statistical environment with unprecedented flexibility and performance [9]. With Biocep, Amazon EC2 instances running R servers can be dynamically launched and terminated to meet load requirements in highly scalable web applications. To illustrate how the deployment side can benefit from open standards and cloud computing, we use the ADAPA scoring engine.

ADAPA is able to generate scores out of a variety of predictive algorithms expressed in PMML. It uses JDM Web Service calls to allow for automatic decisions to be virtually embedded into systems and applications throughout the enterprise. To minimize total cost of ownership, scoring is available as a service through the Amazon EC2 infrastructure. This partnership with Amazon allows for anyone anywhere in the world to deploy and execute predictive models. Through the use of the Amazon Payments system, users can subscribe to the scoring engine in a way very similar to how they would buy books online on the Amazon website (Figure 3).

Figure 3: Amazon Web Services (AWS) provides the virtual data center infrastructure to launch virtual ADAPA instances while the Amazon Payments system precisely tracks usage and performs billing functions for ADAPA customers.

Once an AWS account is set-up [3], the user is granted access to the ADAPA Control Center (ACC). The ACC fulfills three main roles:

1. Authentication: The login process uses the login credentials and password (keys) obtained from Amazon after ADAPA subscription.

2. Management of ADAPA instances: It displays a table containing all the instances that were instantiated together with their current status (initializing, running, shutting-down) and allows for instances to be terminated or rebooted.

3. Launching of new instances: Instances can be launched in a few minutes and are offered in different types, e.g., small, large, and extra-large, aimed to address different processing needs (in terms of processing power, memory and I/O performance) for data processing and model execution.

The process of launching an instance corresponds to the traditional scenario of buying hardware and installing it in a server room. The only difference is that the server in this case sits in the cloud, comes with a pre-installed version of the scoring engine, and launches in just a few minutes, on-demand and ready to be used.

Therefore, an instance comes with no in-house hardware and software to maintain and when no longer necessary, it can be terminated as easily as it was launched. Each instance encompasses a private and secure virtual server running the scoring engine. Once an instance is up and running, it is available to the outside world and can be accessed through its web console [4].

3.1 Model Verification and Execution

Once models are built in one's favorite environment, they can be easily deployed and tested in ADAPA through its web console and be used in batch mode or real-time (Figure 4).

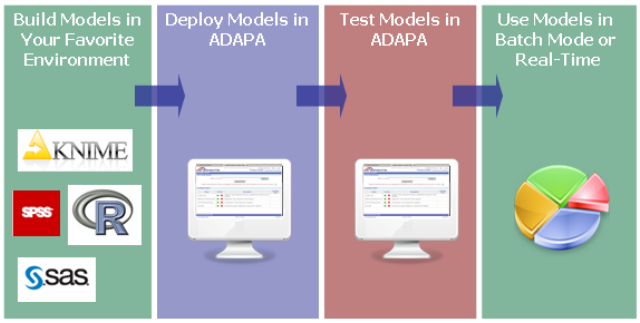

Figure 4: Typical tasks in the life cycle of a data mining project: Building, deploying, testing and using data mining models.

Note that we now operate in a true cross-platform, multi-vendor environment. Since models are developed outside of ADAPA in various, PMML-compliant tools, a prudent step in the deployment process is the model verification, which ensures that both the scoring engine and the model development environment produce exactly the same results. ADAPA provides an integrated testing process to make sure a model was uploaded successfully and works as expected. It allows for a test file containing any number of records with all the necessary input variables and the expected result for each record to be uploaded for score matching.

This model verification process is done through the web console. After processing the file, statistics are returned on total amount of matched and unmatched records and percentages. If any records failed the matching test, a list of failed records is displayed. One can then trace through computed information for each record to locate where expected and computed values differ and thus pinpoint the source of the problem.

[3]The AWS account set-up is a one time event and can be performed in a few minutes.

[4]The web console for a particular instance can be reached simply by clicking on the terminal icon for that instance in the ACC table of instances.

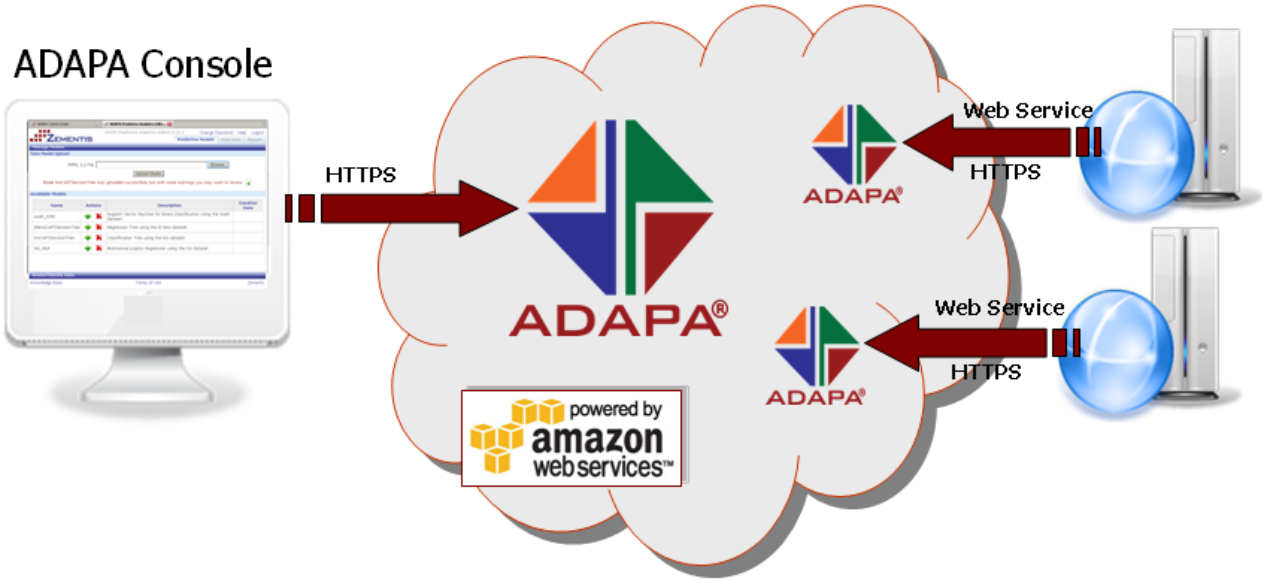

Figure 5: Accessing models in ADAPA through the web console or web service calls.

Typically, once the deployment of a model has been verified, it is deemed ready for execution. As shown in Figure 5, models can be accessed in two different ways:

- Batch execution through the web console: The web console allows for model deployment as well as batch-mode processing of data files. Files can be uploaded in ADAPA containing millions of records and run against any pre-loaded models. When processing is done, the same file plus computed results is available for downloading.
- Real-time execution via web services: Web Service calls can be embedded anywhere in the enterprise. In this way, as soon as data becomes available, it can be placed on a SOAP request and submitted to ADAPA for processing. ADAPA will return a response containing the inputs as well as the results obtained by the model.

Batch execution implies that a file containing many records is uploaded for processing in ADAPA through the web console. Currently, files can be uploaded in CSV format. Files can also be zipped before being uploaded and submitted for scoring. In this case, after processing is completed, ADAPA will make available for downloading a zipped file containing the input and predicted values.

3.2 Security on the Cloud

Allowing a third party service to take custody of proprietary information raises questions about security and control. Besides benefiting from all the Amazon physical and operational security processes [2], security measures were implemented to ensure that data and models are protected with the ADAPA scoring engine. As a decision engine, ADAPA does not need to store any data, it merely acts as a pass-through, receiving requests and returning computed results. When an ADAPA instance is launched on the cloud, nothing is shared. The instance is completely private (only the customer who launches the instance has its access keys). Access to the instance is only granted via HTTPS. The ADAPA console and web service are password protected. In addition, ADAPA utilizes the Amazon EC2 firewall to the extent of maximum lockdown. Once an instance is terminated, all models and data are deleted within seconds, i.e., no residual information is retained.

4. PERFORMANCE

Performance and scalability have been key design principles for ADAPA. To demonstrate this we have measured batch scoring performance for the different Amazon EC2 instance types. These differ in terms of processing power, available memory, and platform (32 or 64 bit). Processing power is expressed in terms of EC2 Compute Units. One unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor. In addition, a certain number of Virtual Cores (VC) is allocated to each type. The Standard Large instance, for example, offers two virtual cores with two EC2 Compute Units each. As mentioned in Section 3, the ACC allows users to launch any type of ADAPA instance to address their needs for data processing and model execution.

For our experiment, we launched one ADAPA instance on the Amazon cloud for each of the available types. We used the "ElNino_NN" model and artificially generated data. Our goal was to measure the processing throughput and scalability under a batch scoring scenario. The results we present here are based on a data file containing 10 million records. Each record was composed of six numeric input fields. The size of the original file is 318 MBs, but it was compressed (zipped) down to 2.4 MBs before it was uploaded to ADAPA. In table 1, we summarize the results for the five different instance types. The first column lists the name of each type. Columns two and three contain their processing power characteristics. The fourth column displays the total time (in seconds) ADAPA took to score all 10 million records. Please

Table 1: ADAPA batch scoring performance (10 million records) for different instance types.

Instance Type	Units	# VC	Time (sec)	rec/sec
Standard Small	1 EC2	1	844	11,848
Standard Large	4 EC2	2	331	30,211
Standard XL	8 EC2	4	174	57,471
High-CPU Med	5 EC2	2	321	31,153
High-CPU XL	25 EC2	8	122	81,967

note that this does not include data upload time, which is the same for all instance types and it is only dependent on the available bandwidth. The last column reflects the corresponding throughput in terms of records processed per second. Figure 6 plots this last column for the different instance types.

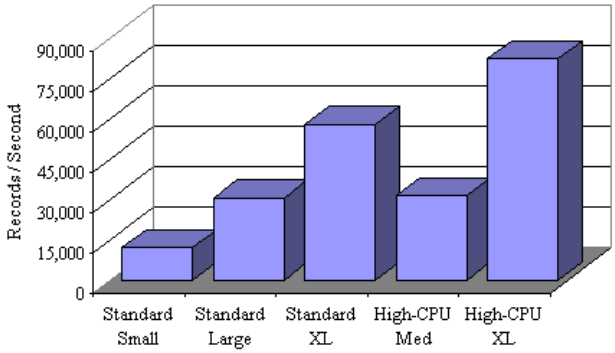

Figure 6: Number of records processed per second for each instance type.

These results show the high levels of throughput that can be achieved with cloud computing and how that scales with available processing power. With the least powerful (and least expensive) instance type - Standard Small, ADAPA can score over 10,000 records per second. Even higher levels of throughput can be obtained by using instances with increased processing capacity. At the high end, it can score 10 million records in about 2 minutes (122 seconds), at a rate of over 80,000 records per second.

Such a high throughput can be sustained for even larger data sets. For example, with a High-CPU XL instance, ADAPA scored 100 million records in 22 minutes. Given ADAPA is available as a service in the cloud, this means that within one hour one can use a model to score as many as 300 hundred million records and pay only a few dollars.

In addition to raw processing performance for scoring data, please note that ADAPA especially accelerates the "speed" of deployment and integration for predictive analytics. While it is possible to scale processing speed with additional hardware, deployment and integration are often the real bottleneck for projects. Only a framework that leverages open standards for interoperability provides the necessary agility required for proper management of predictive models.

5. CONCLUSIONS

Cloud computing offers a powerful and revolutionizing way for putting data mining models to work. It provides a new avenue for science and industry to leverage the power of predictive analytics. We illustrate such a feat using ADAPA, a production-grade software product. Within it, cloud-based virtual machines can be launched on-demand to address the most complex computational tasks.

Predictive models, once expressed in PMML and deployed in the cloud, can be easily accessed from anywhere in the enterprise by the use of web-service calls and managed through a web browser. As a consequence, data mining solutions can now be made available for use in a matter of minutes, not months.

The combination of open standards and cloud computing offers a true revolution in the way data mining models are approached. In addition to accelerating the process of deployment, which allows for businesses to start benefiting right away from their predictive analytics solutions, it also makes it much more affordable. Instead of the traditional costly setup of hardware and software, analytics as a service allows for customers to only pay for resources that they actually consume. Also, by avoiding upfront costs and by being able to express models in PMML, there is no vendor lock-in.

It is our vision for the community that users will be free to share models among many solutions, benefiting from an environment in which interoperability is truly attainable. This vision may come soon to fruition as the leading analytics vendors further commit to improve PMML compliance and invest in educating the industry about the various benefits of the standard [20].

Finally, cloud computing will allow us to lower the cost for data mining and provide a roadmap for predictive analytics to take their place in new applications and industries. The cloud will become a conduit for software and service offerings, making deployment and execution faster and easier, through minimizing the common IT overhead on one side or by providing unprecedented scalability in other cases.

6. REFERENCES

[1] Amazon Web Services. Amazon Elastic Compute Cloud, 2009. http://aws.amazon.com/ec2/.

[2] Amazon Web Services. Amazon Web Services: Overview of Security Processes, 2009. http://aws.amazon.com/.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A Berkeley view of cloud computing, 2009. UC Berkeley RAD Laboratory, http://berkeleyclouds.blogspot.com/.

[4] A. Asuncion and D. Newman. UCI machine learning repository, 2007. University of California, Irvine, School of Information and Computer Sciences. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[5] D. Borthaku. *Hadoop Distributed File System Architecture*, 2008. http://hadoop.apache.org/core.

[6] R. Buyya, C. S. Yeo, and S. Venugopal. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *Proceedings of the Tenth IEEE International Conference on High*

Performance Computing and Communications. IEEE, 2008.

[7] N. Carr. *The Big Switch: Rewiring the World, from Edison to Google*. W. W. Norton and Company, Inc., 2007.

[8] P. Chapman, J. Clinton, R. Kerber, T. Kabaza, T. Reinartz, C. Shearer, and R. Wirth. *CRISP-DM 1.0: Step-by-Step Data Mining Guide*, 2000. http://www.crisp-dm.org/.

[9] K. Chine. Biocep, towards a federative, collaborative, user-centric grid-enabled and cloud-ready computational open platform. In *Proceedings of the Fourth IEEE International Conference on eScience*. IEEE, 2008.

[10] Data Mining Group. PMML Version 3.2, 2009. http://www.dmg.org/pmml-v3-2.html.

[11] T. Erl. *Service-Oriented Architecure (SOA): Concepts, Technology, and Design*. Prentice Hall, 2005.

[12] Expert Group - Specification Lead: M. Hornick. *Java Specification Request 73 (JSR-73): Java Data Mining (JDM)*, 2004. http://www.jcp.org/en/jsr/detail?id=73. Version 1.0.

[13] R. Grossman and Y. Gu. Data mining using high performance data clouds: Experimental studies using sector and sphere. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*. ACM, 2008. http://www.opendatagroup.com/grossman-kdd-08.pdf.

[14] B. Hayes. Cloud computing. *Communications of the ACM*, 51(7):9–11, 2008.

[15] E. Newcomer and G. Lomow. *Understanding SOA With Web Services*. Addison-Wesley, 2004.

[16] R. Nisbet, J. Elder, and G. Miner. *Handbook of Statistical Analysis and Data Mining Applications*. Academic Press, 2009.

[17] J. Taylor and N. Raden. *Smart Enough Systems: How to Deliver Competitive Advantage by Automatic Hidden Decisions*. Prentice Hall, 2007.

[18] G. Williams, M. Hahsler, A. Guazzelli, M. Zeller, W. Lin, H. Ishwaran, U. B. Kogalur, and R. Guha. *pmml: Generate PMML for various models*, 2009. http://cran.r-project.org/package=pmml R package version 1.2.7.

[19] World Wide Web Consortium. W3C Technologies, 2009. http://www.w3.org/#technologies.

[20] M. Zeller, R. Grossman, C. Lingenfelder, M. Berthold, E. Marcade, R. Pechter, M. Hoskins, W. Thompson, and R. Holada. Report on KDD conference 2009 panel discussion: Open standards and cloud computing. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009)*. ACM, 2009.