ABSTRACT
The KDD-2006 Data Mining Standards Workshop included a paper on PMML Conformance, entitled “Conformance Standard for the Predictive Model Markup Language.” That paper was prompted by earlier workshops which identified an issue with the differences in the PMML supported by various vendors. Those differences were seen as an impediment to greater adoption of the standard. It was felt that the lack of conformity had the effect of reducing the usefulness of PMML and hampering the growth of its use by the data mining community [1]. At that time, the standard’s approach to conformance was to specify features that were Core and Non-Core. However, since PMML contains over 700 language elements, it had become impractical for the PMML committee to debate and decide on each and every element. The 2006 paper proposed a different conformance approach to “improve the interoperability of PMML models, and to increase the reliability of PMML as a seamless, multi-vendor model exchange medium.” This paper provides an update on the state of PMML Conformance and the progress made over the past five years. The basis for measuring this progress comes from an analysis of the hundreds of models that have been submitted to the DMG website over the past five years.

Categories and Subject Descriptors

Keywords
Algorithms, Management, Documentation, Standardization, Languages. Data Mining, PMML, Database, Business Intelligence

1. INTRODUCTION
Over the past decade, the Predictive Model Mark-up Language (PMML) has grown to become the de facto standard for documenting and exchanging predictive models [1]. Evidence of this is the number of vendors supporting the standard, which has also grown steadily. While this popularity clearly signifies the value of PMML for vendor specific reasons, the real promise of PMML is its use for cross-vendor interoperability.

At the KDD-2005 Workshop on Data Mining Standards, there was a general consensus that PMML needed to “Get Serious” with respect to model conformance [2]. At the time, increased adoption of PMML came with a variety of interpretations of the PMML specification. “As PMML usage becomes more widespread, the need for some kind of PMML conformance standard has also increased.” Producers of PMML models needed some way to ensure their models are deployed properly. Similarly, consumers of PMML models needed to make sure the PMML they interpret was well formed.

In response to this, in 2006, I wrote a paper that proposed a PMML Conformance approach which would allow PMML producers and consumers to know they can rely on PMML for model interoperability [3]. As an active member of the Data Mining Group (DMG), the consortium behind the PMML standard, it’s been my privilege to work with the other DMG members on implementing improvements in this area. This paper follows-up on the state of PMML Conformance, five years after the 2006 proposal was made.
2. BASIS FOR CONFORMANCE

Data mining is the process of building and deploying models. Therefore, any conformance standard would need to ensure that model producers generate PMML that can be faithfully deployed by model consumers.

First, building models is the job of model producers, and the PMML that describes those models needs to be “valid”. Hence, a conformance standard must include the means for PMML Validation.

Second, those models are deployed to model consumers and, unless that deployment is strictly for archival, reference, or visualization purposes, then scores will be generated from the PMML and those scores must be consistent with the scores the producers expect. Hence, a conformance standard must include the means to verify that models are scored consistently.

Both situations require PMML and data. Producers take data and generate PMML models. Consumers take models and produce data or, more specifically, scores. This relationship is shown schematically in Figure 1.

![Figure 1: Components of PMML Conformance](image)

In order to realize this conformance concept, the 2006 proposal requested that the DMG make the following requirements happen:

1. Provide a page on the DMG web site for PMML Conformance.
2. Provide an interface where producers can submit their PMML for Validation testing. Two options are possible:
   a. Email submission: Good short term solution but requires someone to manually execute XSD and XSLT Validations, and then provide results.
   b. On-line submission: Long term solution that automates the process of PMML Validation testing and reporting of results.
3. Provide XSLT for producer validation, including a process for continuous improvement of the XSLT.
4. Provide an interface on the DMG web site for submitting Model Verification results.
5. Summarize the submitted PMML, including producer and consumer results and an interoperability matrix.
6. Provide information on PMML element usage.

The requirements can be grouped into four areas:
- Conformance Support via Web (#1, #5)
- PMML Validation Utility (#2, #3)
- Model Verification Reports (#4)
- PMML Element Coverage (#6)

The following sections provide a progress report on the achievement for PMML Conformance as compared to these requirements from the original 2006 proposal.

3. CONFORMANCE SUPPORT VIA WEB

In response to this proposal, several improvements were made to the DMG Website [4] to assist with conformance. In particular,
- A wide variety of sample models, built from public domain datasets and generated by a diverse set of vendors, are readily available from the PMML Examples page.
- The Examples page also includes the datasets needed to train and score models.
- The PMML-Powered page lists vendors that support the standard.
- All versions of the PMML standard are available, going back to PMML 1.1.
- A PMML Converter is available that validates and updates models to the latest PMML version. More on PMML Validation in section 4 of this paper.
- An interactive dashboard is available allowing users to explore which PMML elements are used by models validated via the DMG Website. More on PMML Coverage in Section 6.

Overall, most of the PMML Conformance goals established for the DMG Website have been met. The notable exception is Model Verification. Model Verification will be discussed in more detail in Section 5.

GRADE FOR CONFORMANCE SUPPORT: B,

for meeting and/or exceeding requirements in most areas, with the exception of Model Verification (see section 5.0).

4. PMML VALIDATION UTILITY

PMML Validation can be broken down into two parts. First is making sure PMML is syntactically correct with respect to the standard’s XSD. Since tools exist that will validate that an XML document adheres to the specified XSD, one would think that this would be a fool-proof way to determine if a given PMML is valid. And, in many cases, this is true. But unfortunately, certain PMML elements (such as Arrays and Extensions) can cause valid PMML
documents to fail XSD Validation. The PMML standard also contains “meta rules” that are not enforceable via XSD validation (such as the requirement that decision tree evaluation always ends on a “true” node).

Therefore, the conformance proposal recommended using an Extensible Stylesheet Language Transformation (XSLT) validation approach. This approach uses a set of rules that cover particular requirements of the PMML specification. These rules are embodied into XSLT and are applied to a particular PMML using an XSL processor, which will detect any rules that are violated. Using rules that detect invalid PMML use, false negatives (accepting invalid PMML) are avoided. But it was acknowledged that this approach was susceptible to false positives (accepting invalid PMML). Since it’s not practical to create rules that anticipate every possible problem that could occur in PMML, the DMG employed a continuous improvement process [5] where the XSLT would be updated as validation omissions were detected. In this way, over time, the quality of XSLT validation would improve.

Since 2006, the DMG Website has provided a PMML Validation utility where users can submit their PMML for validation. MicroStrategy, employer of this paper’s author, provided the initial XSLT validation code which was maintained by the DMG (since modifying the XSLT was relatively easy to do).

In 2008, a new tool was donated to the DMG by Zementis. The Zementis PMML Converter tool up-converts a PMML document to the latest PMML version [6]. The Zementis PMML Converter has value, since it allows older models to be converted to the newest PMML version, effectively of extending the life of older PMML models (assuming that, over time, vendor support for older versions of PMML will decline). And, scoring engines that depend on the Zementis PMML Converter only have to support the latest version supported by the converter.

For example, a PMML 2.1 model would be converted to PMML 4.0 automatically. Any exceptions would be detected and given a “Fail” result; otherwise, the PMML would be given a “Pass” result.

Just as with the XSLT approach, the Zementis PMML Converter also suffers from the same susceptibility to false positives, (i.e., giving a Pass result to an invalid PMML model). Unfortunately, the Zementis PMML Converter does not participate in the envisioned continuous improvement process for PMML Validation where known false positives are corrected since that tool is only intended for conversion, not full validation.

In fact, an argument can be made that transformations and conversions will always suffer from false positives because they’re based on rules that can never be complete, even with continuous improvement.

Why can’t they be complete?

There are at least two reasons. First, as more vendors generate PMML, there are more opportunities for misinterpretations that need to be detected. And second, each new feature added to the PMML standard requires new features also be added to the PMML validator.

A better way to validate a PMML document is to actually prepare it for scoring. All scoring engines must parse the PMML in order to prepare the execution environment to score new records. This process is typically quite good at detecting problems with the PMML itself.

An even better approach to validate a PMML document is to actually use it to score records. This requires the PMML be valid enough for the scoring engine to properly interpret the model and generate results. At this point, the burden shifts from the model producer generating valid PMML to the model consumer faithfully implementing a scoring engine that can generate verifiable results from valid PMML.

Ultimately, the best PMML validation approach includes an ability to verify the results generated by the model consumer match the results expected by the model producer. Fortunately, PMML includes features that do just this. These features will be discussed in the next section.

**GRADE FOR PMML VALIDATION: C.**

since validating PMML usually works, but false positives can still occur.

5. **MODEL VERIFICATION REPORTS**

It’s the burden of the model producer to ensure that the PMML generated is syntactically and semantically valid. Correspondingly, it’s the burden of the model consumer to faithfully generate scores from that PMML.

In support of this concept, PMML includes a mechanism to ensure the results generated by consumers match those generated by the model producer. PMML’s ModelVerification element allows producers to persist a set of pre-scored model verification records that can be used to verify that the results from the model creation environment match those from the model deployment environment. Each model verification record contains all required input parameters and the corresponding expected result. By comparing the results generated by the model consumer to those persisted in the PMML by the model producer, it’s possible to verify scoring engines are faithfully deploying the predictive model.

Of course, it can be argued that true model verification can only occur if the model verification records cover every possible scoring situation a consumer would need to handle. And, since there’s no practical way to guarantee the amount of the solution space covered by the model verification records, even if model verification succeeds, it could be considered just a sanity check that model deployment is not
broken; necessary but not sufficient to certify a model consumer always generates accurate scores.

But since the model producer controls the composition of the model verification records, the model consumer cannot be held responsible if the coverage of those records is lacking.

Model verification is a benefit for the producer and, more importantly, their users. Producers who take advantage of the ModelVerification feature do so to ensure their models are properly deployed. There’s an implicit assumption that the model verification records are the standard deemed necessary by the producer to verify their model has been deployed correctly.

Therefore, a consumer that correctly scores the model verification records has demonstrated that it faithfully meets the verification standard set by the producer. By providing model verification results, consumers also benefit from Model Verification since they have a reliable way to demonstrate accurate model interoperability.

Of course, there’s no requirement that producers include model verification records, or that model consumers report their model verification results. The 2006 conformance proposal identified two ways to promote the use of model verification records, both utilizing the DMG website:

- Provide an interface on the DMG website for submitting Model Verification results.
- Report producer and consumer results via an interoperability matrix on the DMG website.

Unfortunately, the DMG Website does not include a mechanism for model verification results to be reported and it does not report any model interoperability results. This is an improvement opportunity for the DMG since promoting model verification results is the best way to demonstrate the standard is meeting its goal of enabling cross-vendor predictive model interoperability.

GRADE FOR MODEL VERIFICATION:

Incomplete, since this area has yet to be addressed.

6. PMML ELEMENT COVERAGE

Models submitted to the DMG Website for PMML Validation are also analyzed to determine which PMML language elements are used. The results of this analysis are reported on the DMG website’s model coverage page via an interactive coverage tool. This dashboard allows investigation of which model elements are used by models from different PMML versions and different vendors/applications, representing all the different PMML model types [7]. This feature of the DMG Website is provided by MicroStrategy, the employer of the author.

As of the writing of this paper, 371 unique models submitted to the DMG website were analyzed and these results are summarized here. Duplicate model submissions and models given a Fail result were not included.

As shown in Figure 2, about one third of the submitted models were PMML version 3.1 and another third 3.2. The relatively small number of PMML 4.0 models (the most recent PMML version) indicates that vendors are still migrating to the latest PMML version.

![Figure 2: Submitted Models by PMML Version](image)

Figure 2 shows the breakdown of submitted models by model type. This profile affirms the observation that decision trees, regression, and cluster analysis are the three core algorithms used by data miners [8]. While some model types are clearly less common, all PMML model types are represented.

![Figure 3: Submitted Models by Model Type](image)

Figure 3 shows the breakdown of the submitted models by vendor. Vendors were determined from the PMML Header’s element.
SPSS, acquired by IBM in 2009, supports PMML in most of their products, so it’s not surprising that PMML generated by these popular products are well represented. About a third of the submitted models come from various SPSS products, primarily Statistics and Clementine, including pre-IBM acquisition versions and the re-branded IBM versions of those products (which, for the purpose of this analysis, are still attributed to SPSS for historical reasons).

Models from open source applications, including the Rattle package for R, KNIME and RapidMiner from Rapid-I are also represented.

It’s also important to note that about 20% of the submitted models come from a diverse group of nearly three dozen different applications. This is perhaps the strongest evidence of PMML’s broad adoption and that PMML has truly become a “de facto” standard.

From a conformance point of view, the value of this analysis is to understand which language elements of the PMML standard are in use today. Of the 1,104 unique elements and enumerated attributes in PMML 4.0, 573 or 51.9% are covered by these 371 models. In other words, only about half of the PMML features are utilized in the submitted models.

Of special interest are those language elements that are used in PMML from only one vendor. Appendix A at the end of this paper lists the elements used only by a single vendor. Many of these elements are descriptive (not predictive) and therefore don’t affect model scoring and have limited impact on model interoperability, such as ModelExplanation elements like ANOVA, lift charts and correlation matrices. But more noteworthy is the single vendor support of major predictive elements such as TimeSeriesModel and TextModel, as well as relatively minor features like TreeModel’s missingValueStrategy features “aggregateNodes” and “nullPrediction.”

Of course, this survey is not exhaustive and there very well may be more coverage of PMML elements in models not submitted to the DMG website. And, hopefully continued publication and promotion of the PMML Conformance initiatives mentioned in this paper will result in better and more complete analysis of PMML usage.

Nevertheless, this information can aid model producers who are concerned with cross-vendor interoperability since they can see which elements are widely used and which are not. Similarly, model consumers can focus their efforts on supporting those language elements used by the model types and vendors of interest to them.

**GRADE FOR PMML COVERAGE: A,**

justified by the fact that users can explore the coverage of PMML language features based on models submitted to the DMG website, which was the original goal proposed in the 2006 paper.

7. **Conclusion**

Over the past five years, significant progress has been made towards establishing robust mechanisms for PMML Conformance. Features added to the DMG Website allow users to perform PMML validation and provide insights in the usage of PMML features. But PMML still suffers from the perception that interoperability is limited [9].

As identified above, gaps remain for model validation, model verification and interoperability. In closing, here are six updated requirements that can help to close these gaps:

1. Keep the same user interface -- upload a file and get your results (no registering or log-in required).
2. Promote the use of Model Verification to increase confidence that PMML is a trustworthy solution for predictive model deployment.
3. Expand the PMML Validation approach to include prepping the model for scoring and scoring Model Verification records (when present)
4. Commit to continuous improvement for PMML validation by:
   a. Capturing all reported false positives or false negatives results.
   b. Report known issues that have been fixed.
   c. Report known issues that remain open.
5. The DMG analyzes all validated PMML and report PMML Element Coverage.
6. Report model verification results via the DMG Website as an indicator of interoperability.

8. **Acknowledgements**

The author would like to thank all the colleagues in the industry who commit their valuable time and energy crafting and promoting standards for data mining. Though
too numerous to name here, it has been a privilege working with and learning from the brilliant people who, in addition to their “day jobs,” are able to lend their expertise to these collaborations. I acknowledge that my involvement with the DMG might make some question my objectivity and accordingly I tried to include supporting facts for the more subjective opinions contained in this critique. The achievements noted here are the result of contributions from the dedicated DMG members who volunteer their time and generously share their expertise. I share responsibility for any shortcomings noted; any criticisms are made in the spirit of identifying opportunities for future improvement.

9. References


### Appendix A: Model elements supported by a single vendor

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Model Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroStrategy</td>
<td>Correlations/CorrelationValues</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>Correlations</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>CorrelationValues/Matrix</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>CorrelationValues</td>
</tr>
<tr>
<td>IBM</td>
<td>DataDictionary/Taxonomy</td>
</tr>
<tr>
<td>DSS</td>
<td>DataField[@dataType='dateDaysSince[0]']</td>
</tr>
<tr>
<td>SPSS</td>
<td>DecisionTree[@functionName='classification']</td>
</tr>
<tr>
<td>InfoDecipher</td>
<td>Discretize[@dataType='double']</td>
</tr>
<tr>
<td>Webtrends</td>
<td>DocumentTermMatrix/Matrix</td>
</tr>
<tr>
<td>Webtrends</td>
<td>DocumentTermMatrix</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>ExponentialSmoothing/Level</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>ExponentialSmoothing/Seasonality_ExpoSmooth</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>ExponentialSmoothing/TimeValue</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>ExponentialSmoothing/Trend</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>ExponentialSmoothing</td>
</tr>
<tr>
<td>SPSS</td>
<td>GeneralRegressionModel/ModelVerification</td>
</tr>
<tr>
<td>SPSS</td>
<td>GeneralRegressionModel/PCovMatrix</td>
</tr>
<tr>
<td>SPSS</td>
<td>GeneralRegressionModel[@cumulativeLink='probit']</td>
</tr>
<tr>
<td>SPSS</td>
<td>GeneralRegressionModel[@distribution='gamma']</td>
</tr>
<tr>
<td>SPSS</td>
<td>GeneralRegressionModel[@distribution='negbin']</td>
</tr>
<tr>
<td>SPSS</td>
<td>GeneralRegressionModel[@linkFunction='log']</td>
</tr>
<tr>
<td>SPSS</td>
<td>GeneralRegressionModel[@modelType='generalLinear']</td>
</tr>
<tr>
<td>SPSS</td>
<td>GeneralRegressionModel[@modelType='ordinalMultinomial']</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>LiftData/ModelLiftGraph</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>LiftData/OptimumLiftGraph</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>LiftData</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>LiftGraph/XCoordinates</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>LiftGraph/YCoordinates</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>LiftGraph</td>
</tr>
<tr>
<td>BlueLine</td>
<td>MapValues[@dataType='double']</td>
</tr>
<tr>
<td>IBM</td>
<td>MatCell</td>
</tr>
<tr>
<td>IBM</td>
<td>Matrix/MatCell</td>
</tr>
<tr>
<td>IBM</td>
<td>Matrix[@kind='any']</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>MiningField[@missingValueTreatment='asMode']</td>
</tr>
<tr>
<td>KXEN</td>
<td>MiningModel/LocalTransformations</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>MiningModel/ModelExplanation</td>
</tr>
<tr>
<td>MicroStrategy</td>
<td>MiningModel/ModelStats</td>
</tr>
</tbody>
</table>
KNIME PolynomialKernelType
MicroStrategy PredictiveModelQuality/LiftData
SPSS PredictiveModelQuality[@dataUsage='training']
SAS PredictorTerm/FieldRef
SAS PredictorTerm
Zementis Predictor
SPSS Regression/LocalTransformations
Salford Regression[@normalizationMethod='none']
TreeNet Regression[@normalizationMethod='softmax']
MicroStrategy RegressionModel/ModelExplanation
MicroStrategy RegressionModel/ModelStats
SAS RegressionTable/PredictorTerm
TreeNet Regression
SPSS ResultField[@feature='probability']
MyApplication RuleSelectionMethod[@criterion='weightedMax']
KNIME RuleSet
Rapid-I RuleSetModel/Output
Rapid-I RuleSetModel/Targets
MicroStrategy Seasonality_ExpoSmooth[@type='additive']
MicroStrategy Seasonality_ExpoSmooth
IBM Sequence/Time
IBM SequenceModel/Constraints
MicroStrategy simpleMatching
MyApplication SimpleRule/ScoreDistribution
MicroStrategy SimpleSetPredicate[@booleanOperator='isNotIn']
KNIME SupportVectorMachine
Zementis SupportVectorMachineModel/LocalTransformations
KNIME SupportVectorMachineModel/PolynomialKernelType
KNIME SupportVectorMachineModel/Targets
IBM TableLocator
SPSS TargetValueCount
SPSS TargetValueCounts
IBM Taxonomy/ChildParent
IBM Taxonomy
Webtrends TextCorpus/TextDocument
Webtrends TextCorpus
Webtrends TextDictionary
Webtrends TextDocument
Webtrends TextModel/DocumentTermMatrix
Webtrends TextModel/MiningSchema
Webtrends TextModel/TextCorpus