

A Semantic-Based Analytics Architecture and Its Application to Commodity Pricing

Ali Behnaz¹(✉), Aarthi Natarajan¹, Fethi A. Rabhi¹,
and Maurice Peat²

¹ School of Computer Science and Engineering,
University of New South Wales, Sydney, NSW 2052, Australia
{ali.behnaz, f.rabhi}@unsw.edu.au,
aarthi22@optusnet.com.au

² The University of Sydney Business School, Sydney, NSW 2006, Australia
maurice.peat@sydney.edu.au

Abstract. Over the past decade, several sophisticated analytic techniques such as machine learning, neural networks, and predictive modelling have evolved to enable scientists to derive insights from data. Data Science is characterised by a cycle of model selection, customization and testing, as scientists often do not know the exact goal or expected results beforehand. Existing research efforts which explore maximising automation, reproducibility and interoperability are quite mature and fail to address a third criterion, usability. The main contribution of this paper is to explore the development of more complex semantic data models linked with existing ontologies (e.g. FIBO) that enable the standardisation of data formats as well as meaning and interpretation of data in automated data analysis. A model-driven architecture with the reference model that capture statistical learning requirement is proposed together with a prototype based around a case study in commodity pricing.

Keywords: Ontologies · Semantic · Analytics · Commodity · Statistical learning · FIBO · Architecture · ADAGE · Model-driven engineering · Big data · Data science

1 Introduction

Many areas of science such as geo-sciences, astronomy, genomics and computational physics are confronted with the exponential growth of data. This data presents a vital opportunity to research scientists to understand the behaviour of complex systems and gain fundamental insight. The advent of e-commerce has produced similar growth in economic and business data e.g., security market data, sales forecasts, economic forecasts, inventory studies, workload projection, utility studies, budget analysis, etc. In this paper, we are particularly interested in the analysis of data which consists of observations measured sequentially at discrete points of time, commonly known as time series, e.g., interest rates and exchange rates recorded daily. This data is temporal in nature, it can be modelled deterministically with functions of time and analysed to extract meaningful information that help to better understand the dynamics and

distribution of the data, e.g., draw statistical inferences from the observed data to guide decision making or make predictions about future values of the data based on the previously observed values (forecasting).

Regardless of scientific domain, data analysts are confronted by a number of challenges. Firstly, data analysis is a *complex, time-consuming process* requiring data analysts to combine several independent steps; accessing distributed data sources, local and remote custom software components (e.g. algorithms and scripts) and specialised tools (e.g., statistical tools, mathematical packages) into a larger analysis “pipeline” or process. At a high-level, this pipeline can be divided broadly into four phases: data acquisition, data preparation, data analysis and visualisation [27] (see Fig. 1).



Fig. 1. Different phases in the analytics pipeline

Over the past decade, sophisticated analytic tools and multidisciplinary techniques such as machine learning, neural networks, predictive modelling and data-mining have evolved to enable analysts to derive the needed insights from data [11]. Even though these techniques have emerged as popular strategies for complex analytics, they do not provide an overall solution to analysts conducting *in-sillico* data-intensive analysis. These techniques constitute one element of the overall analysis pipeline, analysts require a broader solution that captures all the phases of data analysis into a “integrated whole”. The primary focus in most research efforts has been the creation of new theories, techniques and software to deal with the complex characteristics of data, practical analytic challenges as perceived by a domain-user have received little or no attention. As data-processing tools and applications are largely developed by software developers, often written in proprietary formats with competing specifications, standards and frameworks the learning curve for domain users is steep and the task of choosing and interacting with the right tools is highly difficult. Data analysis also requires that a data-analyst possess an integrated skill set spanning mathematics, computer-science, machine-learning, artificial-intelligence, statistics and a deep domain knowledge and understanding of the craft of problem formulation [27].

Another challenge that arises in the analysis of temporal data is in the inherent nature of the analytic process itself, which is typically a *computational, quantitative process*. A domain-user typically detects a pattern in the data (e.g., price jump) through computation of measures (e.g., stock return) and then applies these computed variables to several mathematical models and techniques. Analytic dilemma arises in the *planning of the analytic pipeline* when there is a choice of competing *variables* or *measures* that can be computed to detect a similar outcome and the choice of the measure dictates the computation tasks of the analysis pipeline. Translation of the complex computation and analytic model in the minds of the domain user into an analytic process is not a trivial task [27]. Data analysis can also be described as an *exploratory science* characterised by a cycle of model selection, customization and testing as scientists often do

not know the exact goal or expected results beforehand. Data analysts cannot simply look at data and let the data speak for itself. They need to build models to interpret and gain the insight from the data. For example, an investment manager will be relying on a mathematical model to construct an optimal portfolio at a particular point in time. This model will use some underlying time-series variables that represent variations of asset risks and returns over time. The model can be back-tested by “populating variables” with data e.g. historical returns data. Depending on the performance and the accuracy of the model predictions, the user has several options: adjust the mathematical model, change some of the underlying variables or change the way data is mapped into the variables. The entire process is *iterative* in nature, characterised by repeated evaluations on new data-sets or by “tweaking” experimental parameters.

In light of the above challenges, the purpose of this paper is to propose a software architecture that facilitates analytics in a friendly, coherent and technology-agnostic manner. The rest of the paper is structured as follows. Section 2 provides the background and related work. Section 3 presents our solution which is based on a semantic reference model. In Sect. 4, we will apply the proposed model to enable analysts to identify price indicators of commodities. Section 5 concludes this paper.

2 Background and Related Work

Data analytics solutions have evolved from simple analytic techniques, along with supporting analytic tools, to sophisticated problem solving environments for data analysts. There are a wide range of analytic tools, techniques and problem-solving environments at the disposal of analysts. At one end of the spectrum software libraries provide programmers simple building blocks for building sophisticated analysis models and running experiments. There are a multitude of packages and libraries to leverage statistics, machine learning, text-mining, sentiment analysis, etc. [12]. A developer can build a program tailored to their needs utilizing libraries built using programming languages such as Java or Python or use “pre-packaged modules” such as those offered by the R programming language. It has been long argued that many end-users do not have, nor do they wish to, acquire programming skills just to use software packages effectively [13]. Data analysts have valuable skills and should spend their time doing science and not programming or data management. At the other end, application packages such as Microsoft Excel, Google Spreadsheets [20], Gnumeric [19], etc. offer simple, user-friendly interfaces which can be used to conduct elementary to intermediate level analyses. However, there are many shortcomings in such application packages; limited functions, rounding errors, miscalculations, etc. Overall, application packages are very handy tools for building models based on “clean data”, but fail to cater to more sophisticated needs in data analysis.

In this paper, we take a higher level view, looking at high level design i.e. *an architecture* of solutions instead of specific solutions. A complete survey of existing approaches is presented in [11], where it has been noted that service-oriented and scientific workflow based environments are two key architectural approaches that have been extensively used by data scientists in coordinating processes for complex data analysis. *Service-oriented architectures* represent a strategy for the composition of

distributed applications that propagates encapsulation of software artefacts as standards-based services. *Workflow based* approaches model work as a sequence of well-defined steps with a goal of providing a business service or performing a scientific experiment. In this approach each step corresponds to a single unit of work e.g., BPEL [15] and scientific workflow management systems such as Taverna [16], Kepler [17], Wings [18]. These existing approaches have focused on addressing essential criteria to support the activities of analysts, namely (1) automation and reproducibility (2) interoperability [27]. Most architectural strategies have aimed to provide support for a high level of *automation* in order to increase the efficiency and productivity of scientists by helping them to lower the effort and/or reduce time taken to complete tasks. The automation algorithms and code used by scientists contain concise information and instructions that are viewed as an accurate record of the research undertaken. The archived record of these instructions becomes a useful artefact for provenance tracking and ensures reproducibility of analysis processes. To enable the seamless interoperability of diverse tools and packages and enable the execution of process models on multiple platforms, *interoperability* has become another key focus. From this survey, it can be concluded that existing research efforts aimed at maximising automation, reproducibility and interoperability are quite mature, and have helped to mitigate analytic complexity. These approaches still fall short of addressing a third criterion, *usability*, the focus of which is to deliver an analytic approach to enable end-users to work in a more friendly and coherent manner in the face of extreme heterogeneity.

The closest related work to ours is ADAGE [9, 10, 14] which is an architectural framework that aims to achieve user-driven execution of processes through the composition of analysis functions as *services*, providing a reference event data model that enables ADAGE services to process data in a consistent manner. Design concepts underlying ADAGE aim to support data analysts by bringing together existing toolsets and data-sources, but are not adapted to capture the user’s mental models and translate them into analytic pipelines. The ADAGE framework has been extended in [27], where a reference model was proposed to capture an event model and the user’s analytic model with a specific domain. However, the reference model was not implemented using any standard formalisation. The main contribution of this paper is to explore the development of more complex semantic data models linked with existing ontologies that enable the standardisation of data formats as well as meaning and interpretation of data [13].

3 Proposed Approach

To support the design of analytic solutions tailored to the needs of non-technical data analysts we propose an architecture based on Model Driven Software Development and ADAGE principles, at the core of which lies a semantic reference model that represents an abstraction of the computational model to be applied on the data in order to gain insight. In this section, we first provide an overview of the design principles underlying our proposed architecture. Then, we describe semantic and statistical learning concepts. Finally, we present our ontology for modelling statistical learning algorithms.

3.1 Overview of Proposed Architecture

The proposed architecture can be seen as an application of the principles of *Domain Driven Design (DDD)* [21] and *Model Driven Software Development (MDSD)* [22] which emphasise that the heart of software development is knowledge of subject matter or *domain*. DDD is a design approach introduced by Eric Evans which focuses on creating a *model of the domain*, rather than the technology, using a high level of abstraction. This model should not just be a data schema or class diagram, but represents distilled knowledge about the domain and accurately expresses the behaviour of the domain through identifying important domain concepts and relationships between these concepts. The model shows how the domain users think of their domain problems in terms of these concepts. A domain model articulates domain problems and provides a practical approach to software design. It is equally important that the domain model is crafted carefully to enable it to be translated to practical implementations. A domain-driven design naturally leads to a model driven software development approach, which provides a software development approach to realise software systems from domain models. MDSD is currently a highly regarded software development paradigm, and a good fit for our proposed approach because of its “consumer-centric” or “end-user-centric” focus. MDSD approaches a software solution from the domain perspective, specifying that needed application functionality and behaviour be modelled formally in terms of the problem domain, without tainting it with technological concerns. MDSD focuses on *models* as central artefacts, where a model is a formal platform-independent module (PIM) that provides an abstract representation of a real-world application and applies *model transformations* to realise software systems from these PIMs. In our architecture, these models are defined as an ontology, captured using an appropriate semantic technology such as OWL [23], FIBO [24] etc.

Our proposed architecture defines an analytic stack that comprises of four main tiers, depicted in Fig. 2. At the lowest level of the stack is the *data* tier, which comprises different types of raw data to be analysed, such as structured and semi-structured data. The next tier is the *semantic reference model*, which is represented using an appropriate semantic technology such as OWL [23]. The semantic reference model encapsulates an *event model* and an *analytic model*. While the event model provides a standardisation of the representation of data from heterogeneous data-sources within a target domain, the analytic model captures the complex computation model in the minds of the domain user that is to be applied on the domain data. The tier above is the *analytics platform* layer, which encompasses the analytic tools that can be used to apply the analytic techniques embodied in the semantic reference model tier. The top tier is the higher-level *application layer* which can range from simple user interfaces and custom applications that produce visualisation results by running tools in the layer beneath to more sophisticated platforms such as scientific workflow management systems, which orchestrate a pipeline of analytic tasks to compose a single analytic process.

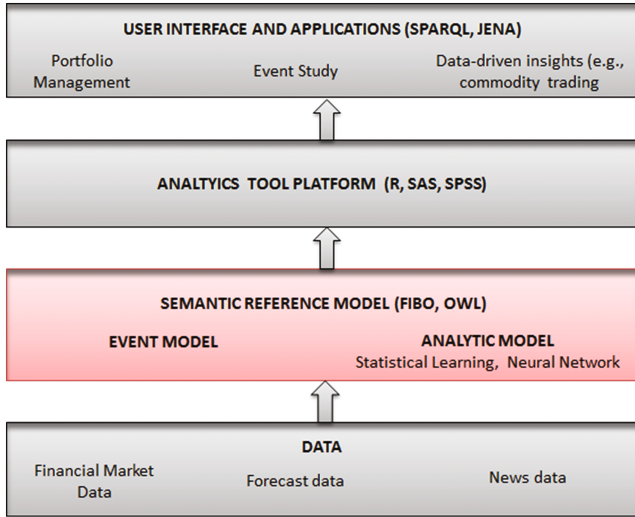


Fig. 2. Reference architecture

3.2 Semantic Reference Model: Basic Principles

The word *semantic* stands for ‘meaning’; a *semantic concept* is a name used by the domain user to identify a domain object within a specific domain. For example stock price is a domain concept in the context of a financial application. *Ontology* describes a body of knowledge about a specific domain by defining the semantic concepts and semantic relationships between these concepts. Semantic relationships model the behaviour of the domain by capturing different kinds of associations between semantic concepts. To provide a formalisation to the vocabularies used in defining an ontology, W3C offers a large range of standard formats such as RDF and RDF schemas, Web Ontology Language (OWL) etc. For example, RDF represents information about the domain as *triples* which are a tuple of the form $\langle \text{Subject}, \text{Predicate}, \text{Object} \rangle$, where *subject* and *object* represent two domain *semantic concepts* and *predicate* is a *semantic relationship* between these resources [28].

According to OASIS (Organization for the Advancement of Structured Information Standards) a *reference model* is:

“an abstract framework for understanding significant relationships among the entities of some environment, and for the development of consistent standards or specifications supporting that environment. A reference model is based on a small number of unifying concepts and may be used as a basis for education and explaining standards to a non-specialist. A reference model is not directly tied to any standards, technologies or other concrete implementation details, but it does seek to provide a common semantics that can be used unambiguously across and between different implementations [2].”

The *semantic reference model* constitutes the core of our analytic stack. As previously noted, a vast amount of research and development has been directed towards analysing heterogeneous datasets, applying fragmented analytic techniques and building

analytic models to unravel patterns. However, no generic model has been proposed to link the scattered knowledge in the data and the computational model in the minds of the domain-users. In proposing a semantic reference model we try to fill a major methodological gap, defining the semantics of a complex analytics model. As stated earlier, the semantic reference model comprises an *event model* and an *analytic model*. This paper focuses on defining an ontology to represent the complex analytic model and will rely on the event model proposed in [26]. Further, for the purposes of this paper, we limit the scope of our semantic analytic model to the analysis of data using statistical learning methods. The analytic model can be easily extended in the future to support other analytic techniques. Statistical learning addresses the general problem of function estimation based on empirical data, encompassing a wide array of popular algorithms and techniques for data analysis, pattern recognition and prediction [8].

Before we define our ontology for modelling statistical learning methods, we also define what statistical learning is using a simple example, as described by [25]. Suppose a researcher is interested to determine if the % change in inflation and the increase in population have an effect on beef consumption. In this context, the % inflation change and population increase are *independent variables* or *predictors* while beef consumption is the *dependent variable* or *response*. If the predictors are denoted as X_1, X_2, \dots, X_n and the response is denoted as Y and Y is affected by the predictors, then we can define $Y = f(X)$, where f is the function that connects the predictors X_1, X_2, \dots, X_n to the response Y . This function, f , is generally unknown and one must estimate f based on observed data points. Statistical learning is a set of methods for estimating this function f . The two primary reasons for estimating f are *prediction* and *inference*. *Prediction* is about the using the estimated function f on a set of predictors, X , to calculate a predicted value for Y . *Inference* is concerned with how the response Y is affected as the predictors $\{X_1, X_2, \dots, X_n\}$ change. There are many linear and non-linear methods for estimating f and these methods can be broadly categorised as *parametric* and *non-parametric* methods [25]. We briefly provide an overview of parametric methods.

Given a set of data points or observations, these observations are referred to as *training data* as these observations will be used to train the method selected to estimate f . A parametric approach involves a two-step model based approach [25].

1. First, an assumption is made about the functional form or model of f , if f is linear in X it can be defined as:¹

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \dots + \beta_p X_p.$$

2. Once a model has been selected, the next step is to *fit* or *train* the model. In the previous step, if a linear model has been chosen the model estimator simply needs to estimate the parameters $\beta_0, \beta_1, \beta_2, \dots, \beta_p$, once values of these parameters have been estimated the function f is defined as,

¹ Function is multiple linear regression, which is a widely used form in statistical learning.

$$Y = f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \dots + \beta_p X_p.$$

One possible and quite commonly used approach to fitting the linear model is referred to as *ordinary least squares*. For our example, once the parameters have been estimated, we have a fitted linear model of the form:

$$\text{beef consumption} = \beta_0 + \beta_1 * (\% \text{ inflation change}) + \beta_2 * (\% \text{ population change})$$

3.3 A Semantic Ontology for Modelling Statistical Learning

We now define our reference model which represents the key entities in statistical learning based on a parametric method. Tables 1 and 2 respectively define key semantic Classes and Properties used in the ontology.

This reference model is compatible with the Financial Industry Business Ontology (FIBO) [3]. The Financial Industry Business Ontology (FIBO) is a business conceptual ontology developed by the members of the Enterprise Data Management Council (EDM Council). FIBO provides a description of the structure and contractual obligations of financial instruments, legal entities and financial processes. FIBO is expressed

Table 1. Semantic classes for statistical learning ontology

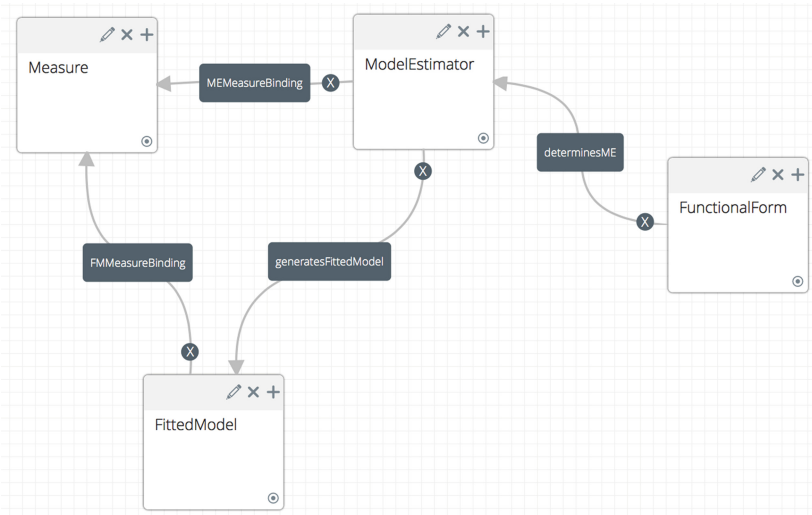
Semantic ontology for statistical learning	
Semantic classes	Description
Functional form	The first step in a parametric based method is to assume a functional form or model for the function f e.g., a linear model or a non-linear model such as thin-plate spline
Measure	A variable e.g. beef consumption which could either be a predictor (independent variable) or a response (dependent variable)
Model estimator	A technique which given some measures, produces a function capable of predicting the values of one measure (dependent variable) based on the value of other measures (independent variables). So for our case study, the model estimator used is shown below e.g., Ordinary Least Squares estimator used to estimate parameters $\beta_0, \beta_1, \beta_2$
Fitted model	The fitted model obtained after the parameters have been estimated using an appropriate model estimator e.g., for our example, after applying the least squares estimator, $\text{beef consumption} = \beta_0 + \beta_1 * (\% \text{ inflation change}) + \beta_2 * (\% \text{ population change})$ and if $\beta_0, \beta_1, \beta_2 = 2, 3, 4$ respectively then $\text{beef consumption} = 2 + 3 * (\% \text{ inflation change}) + 4 * (\% \text{ population change})$ It can be seen for a given functional form assumed (e.g. Linear Model), different kinds of estimators (Least Squares, Lasso, etc.) can yield different fitted models
Function	Represents general functions such as $2 + 3x + 4y$ in the example above.

Table 2. Semantic properties for statistical learning ontology

Semantic properties	Description
generatesFittedModel	Definition: a predicate indicating any fitted model that is generated from model estimator by applying measures (training sets)
FMMeasureBinding (functional Model measure binding)	Definition: a predicate indicating the link between fitted model and the measure(s) in the fitted model.
MEMeasureBinding (model estimator measure binding)	Definition: a predicate indicating the link between model estimator and the measure(s) used it to estimate parameters.
determinesME	Definition: a predicate of Functional Form which indicates the form that is used for predicting a measure.
useFunction	Definition: a predicate that uses any function in mathematical definition.

in the RDF language of the Web (RDF/OWL) for machine readable interface processing and UML for human reading [3].

As shown in Fig. 3 our proposed *Reference Model* has five key classes: *Measure*, *FunctionalForm*, *Function*, *ModelEstimator* and *FittedModel*. These five classes together with ontologies that are part of FIBO capture key concepts of the reference model. The main concepts borrowed from FIBO include the entity *Measure* which represents an amount or degree of something; the dimensions, capacity or amount of something ascertained by measuring [5]. *Measure* is a subclass of the *Reference* ontology in FIBO. *Reference* is a concept that refers to (or stands in for) another concept. Every *Measure* is also a subclass of a FIBO *Thing* [4] which is a set theory construct.

**Fig. 3.** Reference model in Jalapeno

The semantic reference model is implemented using the CAPSICUM framework, which provides meta-models, methods and tooling for developing dynamic, interactive business blueprints [6]. Users are able to maintain the reference model using Jalapeno, which is an interactive modelling platform for building CAPSICUM models. The models can be exported in a variety of formats (e.g. Turtle, RDF, XML, JSON, Marklogic Entity Model). FIBO ontologies have been imported into Jalapeno and are integrated in our reference model.

4 Commodity Pricing Case Study

This section describes a case study in which we explain the context (commodity pricing) and associated reference model, then describe a prototype implementation built following MDE principles.

4.1 Business Area and Associated Reference Model

The case study was inspired by a Hackathon organised at University of New South Wales in partnership with ANZ Bank in Australia. The motivation is that the future success of agribusiness will be reliant on informed decisions about capacity, investment and other driving factors. Many of the banks' customers are interested in questions like "which countries and consumers will buy our products? what prices and economic value is likely to be generated from this? what primary or processed food products should Australia seek to produce in future?". The idea of the competition was to use public and private data on this sector – macro-economic indicators, production volumes, weather patterns, prices, etc. to investigate what will drive this industry going forward [1].

Based on the available data, an instance of the analytics reference model was created to allow heterogeneous datasets to be analysed. Table 3 shows a sample of the measures used in the case study, the reference model would allow thousands of such measures to be defined (Table 4).

For example, applying a functional form "Multiple linear regression" to the measure *Beef and Veal export* (dependent variable) and the measures *Export of goods* and *Employment in agriculture* (independent variables) would produce a linear function of the form:

$$\begin{aligned} \text{Beef and Veal export} &= F(\text{Export of goods}, \text{Employment in agriculture}) \\ &= \beta_0 + \beta_1 \text{Export of goods} + \beta_2 \text{Employment in agriculture} \end{aligned}$$

We have restricted ourselves to regression model estimators, so the models produced are regression functions (characterised by regression factors). Each of the models produced by a model estimator is an instance of a *FittedModel*. Given that there are potentially thousands of measures, it is possible to create millions of models each predicting a particular measure as a function of other measures. The measures themselves are connected (via FIBO ontologies) to other entities. For example, the Measure

Table 3. A sample of measures used in the case study

Measures
China - Beef and Veal export (KT)
China - Beef and Veal import (KT)
China - Beef and Veal Global Consumption
Beef price (us cents per pound)
China - Beef and veal global production (KT)
China - GDP per capita (Yuan)
China - Population (millions)
China - Import of goods (% change)
China - Export of goods (% change)
China - Inflation change (% change)
China - Gross national saving (% of GDP)
China - Per capita beef sold by rural household (kg)
China - Natural growth rate
China - Unemployment rate
Brent crude oil spot rate (USD per barrel)
China - Employment in agriculture (% of total employment)
China - Agricultural land (% of land area)
China - Urban population (% of total)

Table 4. Functional forms used in the case study

Functional form
Multiple linear regression
Multiple exponential regression
Multiple polynomial regression (second degree)
Stepwise regression

“China Beef and Veal Export” is linked to entities “Export” (FIBO Thing), “Beef and Veal” (FIBO EconomicResource) and “China” (FIBO Country). These are represented outside the scope of our case study but can be imported to enable more sophisticated usage of the reference model.

4.2 Prototype Implementation

Based on the general architecture presented in Sect. 3, we have built an analytics tool for identifying price indicators of commodities. The structure of the prototype is illustrated in Fig. 4. The tool has been developed in R and built using libraries such as Shiny and ShinyBS for the User Interface, MASS library to perform stepwise regression, Quandl to get data from quandl.com and XLConnect to read local dataset saved as excel files [28].

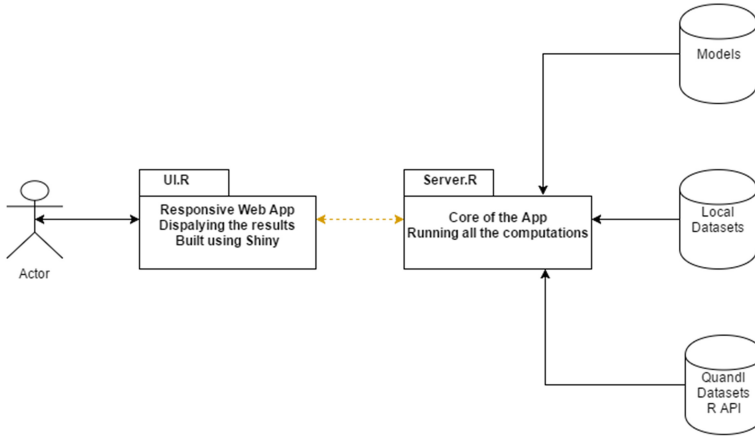


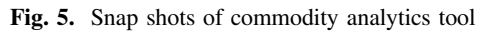
Fig. 4. Structure of the prototype

The tool works in two steps:

- A modelling step: where the user selects a variable to predict (dependent variable) and several possible explanatory variables (independent variables) in order to find the regression equation (model's equation) that describes the variations of the dependent variable.
- A forecasting step: using the equation of the first step and her own views on the independent variables, the user can forecast the dependent variable by inputting values for each of the independent variables which are fed into the model's equation to find a predicted value of the dependant variable.

In Fig. 5 we have provided a snapshot of the tool. The user interface is designed to enhance user interaction, we have grouped the measures by country and commodity. We have also provided an option for selecting models (or Model Estimator) to deploy an analytics model. The tool is equipped with a predictive section which uses the outcome of the analytics model to generate different scenarios. To analyse scenarios, the user can tweak the tolerance of the measures (Forecast Parameters) and select the type of forecasting model.

The structure of the user interface in the model leverages our Semantic Reference Model. All measures shown to users are the result of querying the reference model. In addition, R code is automatically generated from a user query. For example, the snippet below shows generic code in R that implements multiple linear regression once the user has selected the dependent and independent variables.



Scalability is a property of this tool. Additional datasets can be added by modelling the appropriate measures in the reference model and such measures will be immediately available to the user via the User Interface. This architecture allows the user to create and add more analytics models or model estimators.

This paper proposes a model-driven architecture that empowers domain experts to control and guide analytics processes in an exploratory way in the face of heterogeneity and complexity. The focus of this paper is a reference model which encompasses two main features: (1) a semantic model that captures the concepts in statistical learning algorithms and explicitly defines the relationships between variables, functional forms and model estimators, (2) leveraging statistical learning packages (e.g. R), semantic

technologies (e.g. RDF) and existing ontologies (e.g. FIBO) in an innovative fashion to facilitate predictive modelling and automatic code generation.

The work presented in the paper is still in its early stages. Future work will concentrate on three areas. Firstly, the reference model will be generalized so that other analytics techniques such as text processing, sentiment analysis etc. can be incorporated. An important part of this work will include modelling parametric and non-parametric statistical learning techniques. Secondly, the reference model has to take into consideration how the measures are linked to the raw data. For this part, we intend to create an event ontology based on the work in [26]. Finally, we need to leverage all the constructs offered by semantic ontologies, like the ability to make inferences. In particular, the FIBO Relation ontology defines a rich set of relationships between measures that could be exploited, such as the relationship “isCausedBy” to indicate cause-effect relationships. For example, if a measure A “isCausedBy” measure B, and measure B “isCausedBy” measure C, it can be inferred that measure C “isCausedBy” measure A although this is not defined explicitly in the model.

Acknowledgements. We are grateful to ANZ Bank Agribusiness unit, especially Richard Schroder and Felipe Flores, Thomson Reuters and IBM for sponsoring the Hackathon which provided the data for the case study of this paper. We are also grateful to Terry Roach and Max Gillmore for helping on different aspects of this work.

References

1. Info Package for UNSW Data Science Hackathon. http://www.cse.unsw.edu.au/~fethir/HackathonInfo/HackathonStudentPack_v7.pdf. Accessed on 10 Sep 2016
2. OASIS SOA Reference Model Technical Committee. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm/. Accessed on 10 Sep 2016
3. Financial Industry Business Ontology Foundations, The Enterprise Data Management Council. <http://www.edmcouncil.org/edmcouncil>. Accessed on 10 Sep 2016
4. Financial Industry Business Ontology (FIBO), Object Management Group. <http://www.omg.org/spec/EDMC-FIBO/>. Accessed on 10 Sep 2016
5. Merriam Webster, Measure (Definition). <http://www.merriam-webster.com/dictionary/measure>. Accessed on 10 Sep 2016
6. Roach, T.M.: CAPSICUM – A Semantic Framework for Strategically Aligned Business Architecture. Ph.D Thesis, UNSW, Sydney, Australia (2011)
7. Behnaz, A., Rabhi, F., Peat, M.: A software architecture for enabling time series analysis on real-time event data. In: Proceedings of International Work-Conference on Time Series, June 2016
8. Vapnik, V.: The Nature of Statistical Learning Theory, 2nd edn. Springer, New York (1999)
9. Rabhi, F.A., Yao, L., Guabtni, A.: ADAGE: a framework for supporting user-driven ad-hoc data analysis processes. Computing **94**(6), 489–519 (2012). doi:[10.1007/s00607-012-0193-0](https://doi.org/10.1007/s00607-012-0193-0)
10. Yao, L., Rabhi, F.A.: Building architectures for data-intensive science using the adage framework. Concurrency Comput. Pract. Exp. **27**(5), 1188–1206 (2015)
11. Chen, J., Choudhary, A., Feldman, S., Hendrickson, B., Johnson, C., Mount, R., Sarkar, V., White, V., Williams, D.: Synergistic challenges in data-intensive science and exascale computing. DOE ASCAC Data Subcommittee Report, Department of Energy Office of Science (2013)

12. Yao, L., Rabhi, F., Peat, M.: Supporting data-intensive analysis processes: a review of enabling technologies and trends. In: Ramanathan, R., Raja, K. (eds.) *Handbook of Research on Architectural Trends in Service-Driven Computing*, vol. 2, pp. 481–508. IGI Global, Hershey (2014). doi:[10.4018/978-1-4666-6178-3](https://doi.org/10.4018/978-1-4666-6178-3)
13. Bernstien, P.A., Wecker, D., Krishnamurthy, A., Manocha, D., Gardner, J., Kolker, N., Reschke, C., Stombaugh, J., Vagata, P., Stewart, E.: Technology and data-intensive science in the beginning of the 21st century. *Omics: J. Integr. Biol.* **15**, 203–207 (2011)
14. Yao, L.: *ADAGE A Framework For Supporting User-Driven Ad Hoc Data Analysis Processes*. Doctor of Philosophy, University of New South Wales (2013)
15. OASIS, OASIS Web Services Business Process Execution Language (WSBPTEL) TC | OASIS. <https://www.oasis-open.org/committees/wsbpel/>. Accessed 9 Sep 2016
16. TAVERNA 2009, Taverna - open source and domain independent Workflow Management System (2009). <http://www.taverna.org.uk/>. Accessed 9 Sep 2016
17. Tao, J., Zhao, Y.: Scientific workflow management and the Kepler system. *Concurrency Comput. Pract. Exp.* **18**, 1039–1065 (2006)
18. Deelman, E., Moody, J., Kim, J., Ratnakar, V., Gil, Y., Gonzalez-Calero, P.A., Groth, P.: Wings: intelligent workflow-based design of computational experiments. *IEEE Intell. Syst.* **26**(1), 62–72 (2011)
19. Gnumeric.org., Gnumeric (2016). <http://www.gnumeric.org/>. Accessed 17 Sep 2016
20. Apps.google.com. Google Sheets – Spreadsheets & Data Analysis for Business (2016). https://apps.google.com/intx/en_au/products/sheets/. Accessed 17 Sep 2016
21. Evans, E.: *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Professional, Reading (2004)
22. Völter, M., Stahl, T., Bettin, J., Haase, A., Helsen, S.: *Model-Driven Software Development: Technology, Engineering, Management*. John Wiley & Sons, Hoboken (2013)
23. W3.org. OWL Web Ontology Language Guide. (2016). <https://www.w3.org/TR/owl-guide/>. Accessed 17 Sep 2016
24. W3.org. Financial Industry Business Ontology Community Group (2016). <https://www.w3.org/community/fibo/>. Accessed 17 Sep 2016
25. James, G., Witten, D., Hastie, T., Tibshirani, R.: *An Introduction to Statistical Learning with Applications in R*. Springer, New York (2013)
26. Milosevic, Z., Chen, W., Berry, A., Rabhi, F.A.: An open architecture for event-based analytics. Accepted in *Int. J. Data Sci. Anal.* (2016)
27. Natarajan, A.: *Aventis, An architecture for event data analysis*. Doctor of Philosophy, University of New South Wales (2016)
28. Behnaz, A., Rabhi, F., Peat, M.: *A Software Architecture for Enabling Statistical Learning on Big Data*. Springer Series on Statistics (2016)

Enterprise Applications, Markets and Services in the
Finance Industry

8th International Workshop, FinanceCom 2016,

Frankfurt, Germany, December 8, 2016, Revised Papers

Feuerriegel, S.; Neumann, D. (Eds.)

2017, XI, 125 p. 26 illus., Softcover

ISBN: 978-3-319-52763-5