HHS Public Access

Author manuscript

Int J Big Data. Author manuscript; available in PMC 2018 April 13.

Published in final edited form as:

Int J Big Data. 2015 October ; 2(2): 43–56.

Automated Predictive Big Data Analytics Using Ontology Based Semantics

Mustafa V. Nural[*], **Michael E. Cotterell**[*], **Hao Peng**[*], **Rui Xie**[†], **Ping Ma**[†,*], and **John A. Miller**[*]

[*]Department of Computer Science, Statistics University of Georgia, Athens

[†]Department of Statistics University of Georgia, Athens

Abstract

Predictive analytics in the big data era is taking on an ever increasingly important role. Issues related to choice on modeling technique, estimation procedure (or algorithm) and efficient execution can present significant challenges. For example, selection of appropriate and optimal models for big data analytics often requires careful investigation and considerable expertise which might not always be readily available. In this paper, we propose to use semantic technology to assist data analysts and data scientists in selecting appropriate modeling techniques and building specific models as well as the rationale for the techniques and models selected. To formally describe the modeling techniques, models and results, we developed the Analytics Ontology that supports inferencing for semi-automated model selection. The SCALATION framework, which currently supports over thirty modeling techniques for predictive big data analytics is used as a testbed for evaluating the use of semantic technology.

Keywords

semantics; ontology; model-selection; big-data-analytics

1 Introduction

Predictive big data analytics relies on decades worth of progress made in Statistics and Machine Learning. Several frameworks are under development to support data analytics on large data sets. Included in the group are Drill[1], Hadoop[2], Mahout[3], Storm[4], Spark[5], and SCALATION (Miller, Han, & Hybinette, 2010). These frameworks target large data sets by using databases and distributed file systems as well as parallel and distributed processing to speed up computation and support a greater volume of data. As large amounts of data become readily available, one would expect greater use of such frameworks, even by scientists, engineers and business analysts that may not be familiar with the state-of-the-art in Statistics and Machine Learning.

[11]Akka Framework: http://akka.io/

[1]Drill: http://drill.apache.org/
[2]Hadoop: http://hadoop.apache.org/
[3]Mahout: http://mahout.apache.org/
[4]Storm: https://storm.apache.org
[5]Spark: https://spark.apache.org

The rapidly growing need for more people to analyze, or more importantly, make sense of, ever increasing amounts of data is an important challenge that needs to be addressed. One way to address this challenge is more education. Many universities are adding academic and professional programs on data analytics and data science. In addition to this, technology can also help address the problem. As there are over one hundred popular modeling techniques in the fields of Statistics and Machine Learning (SCALATION already supports over thirty), how can one decide? Furthermore, given a modeling technique (type of model), there is still much work left to build a model, including use of data transformation functions, choices of predictor variables, etc.

With smaller data sets and high expertise on the part of the analyst, one practice is to try all possible models for a set of preferred techniques. For big data and less experienced analysts, this practice cannot always be relied upon.

We propose to use Semantic Web technology to assist analysts in selecting, building and explaining models. Statistical and Machine Learning models are formally described using the Analytics Ontology. It is defined using the Web Ontology Language (OWL)[6] and built using the Protégé Ontology Editor and Framework[7]. Its taxonomy (class hierarchy) of model types, equivalence axioms between the model types in the taxonomy and property restrictions can be used to help choose appropriate modeling techniques using a Description Logic (DL) reasoner. This paper focuses on the use of the SCALATION framework and semantic technology to assist in the development and execution of large-scale models.

The rest of this paper is organized as follows: Section 2 discusses the workflow that we use to oversee the entire analytics process. Related work, on model selection and the use of semantics in analytics, is presented in Section 3. Section 4 provides an overview of the SCALATION Framework. Extraction of metadata is presented in Section 5. The structure and design of the Analytics Ontology as well as how this ontology is used in our analytics process is presented in Section 6. Finally, Section 7 concludes the paper.

2 Predictive Analytics Workflow

Abstractly, a univariate predictive model can generally be formulated using a prediction function f as follows:

$$y = f(\boldsymbol{x}, t; \boldsymbol{b}) + \varepsilon$$

where y is the response variable, $\boldsymbol{x}$ is a vector of predictor variables, t is a time variable, $\boldsymbol{b}$ is a vector matrix of parameters and $\boldsymbol{\varepsilon}$ represents the residuals (what the model does not account for). The objective is to pick functional forms and then estimate the parameters, $\boldsymbol{b}$ to the data to in order to minimize the residuals. Estimation procedures for doing this include the following (Godambe, 1991): Ordinary Least Squares (OLS), Weighted Least Squares (WLS), Maximum Likelihood Estimation (MLE), Quasi-Maximum Likelihood Estimation

[6]OWL: http://www.w3.org/TR/owl2-overview/
[7]Protégé: http://protege.stanford.edu/

(QMLE), Method of Moments (MoM) and Expectation Maximization (EM). Closely related to the prediction problem is the classification problem. Although some view classification as a special case of prediction, classification takes center stage when the response variable y takes on values from a set small enough for its elements to be named (e.g., reject, weak reject, neutral, weak accept, accept).

Predictive big data analytics involves many complex steps, many of which require a high-level of expertise. To help manage the complexity, we developed a hierarchical workflow for the predictive analytics process. The top level of our hierarchical workflow is illustrated in Figure 1.

The first step is to load the dataset in a flexible data structure that will make common manipulation operations easy. SCALATION contains a prototype columnar relational database (Stonebraker et al., 2005) that supports relational algebra operations (Codd, 1970). The columns in the relational database are made of vectors making it easy and efficient to perform analytics on data stored in the database.

The second step involves extracting key information regarding the dataset from data and metadata. This information plays an essential role during model type selection phase in addition to the explanation phase described later. Data & metadata extraction is discussed in more detail in Section 5.

The third step in the workflow involves further refinement of the selection problem as well as preparation of the data for analysis. This step involves the handling of missing values, multicollinearity and sparsity of data. It also involves the preprocessing of the various columns in a dataset so that they are encoded numerically. Certain other considerations may be taken at this point as well. For example, if the number of predictor variables is high or the multicollinearity exists, some form of data/dimensionality reduction should be considered (e.g., Regression through Singular Value Decomposition (SVD) or Principal Component Analysis (PCA)).

The fourth step in the workflow, and the main focus of this paper, is the selection of practical modeling techniques or model types based on the characteristics of a dataset. This is performed by first identifying the domains of discourse for each column in the dataset, then applying semantic inferencing using an ontology to suggest potential model types. For example, if the response variable is binary or dichotomous (e.g., grant loan or deny loan), then Logistic Regression becomes a candidate since the residuals are distributed according to a Bernoulli distribution. If the response variable represents counts, *Poisson Regression* should be considered. The characteristics of domains of the predictor variables can be used in selecting between different model types including but not limited to Analysis of Variance (ANOVA), Analysis of Covariance (ANCOVA) and Multiple Linear Regression (MLR).

The fifth step is a diagnostic analysis of the full models (all predictors included) in order to validate the chosen model(s) and go back to the first step if a model(s) does not conform to the assumptions and requirements of the model type and therefore needs to be refined. One of the most important diagnostics is residual analysis. For example, if the first step suggested a General Linear Model (GLM), but the residuals are found not to be sufficiently normally

distributed with zero mean and constant variance, then various stabilizing transformations such as *square-root, log, or other Box-Cox family of transformations* may be considered. Additionally, the distribution of the residuals (e.g., Multinomial) can suggest a different model type such as Multinomial Logistic Regression. Finally, more than one model type may be suggested in the previous step. Diagnostic analysis also involves comparison of models in such cases and choosing the most suitable one according to the metrics provided.

The sixth step in the workflow performs various model building and model reduction techniques to suggest the subset of predictor variables that are used for each model type. For example, in regression the following procedures may be used: Stepwise Regression, Forward Stagewise, Least Angle Regression (LARS) (Efron, Hastie, Johnstone, & Tibshirani, 2004).

Finally, once a model is validated and finalized, it can be used for prediction. Additionally, the results may be explained based on the information available to the system. Information sources include both the metadata extracted from the dataset and the information gathered from external sources such as domain ontologies. In-depth discussion on this topic is provided in later sections.

3 Related Work

Semi-automated/automated model selection has been studied extensively in the literature. Similar to our approach, (Bernstein, Hill, & Provost, 2002) describes an ontology based Intelligent Discovery Assistant (IDA). After analyzing an input dataset, the system generates all possible workflows from the ontology that are valid within the characteristics of the input. The recommendations are then ranked based on user specified criteria (e.g., simplicity, performance, etc.).

A more widely adopted technique used in automated model selection for classification is called metalearning. Metalearning treats the selection problem itself as a learning problem. A metalearning model uses characteristic properties of a dataset and performance metrics of different classification algorithms on that dataset as a training sample and learns to pick the most suitable model for a given dataset from this training exercise. The dataset properties used in metalearning (i.e., meta-features) include simple properties such as number of attributes, sample size, number of classes, and, in most cases, more complex properties such as noise-signal ratio, normalized class entropy, skewness, etc. A comprehensive list of meta-features used in metalearning is provided in (Reif, Shafait, Goldstein, Breuel, & Dengel, 2012). Despite the fact that the field of metalearning is well established, there are only a few studies that focus on applying metalearning for model selection for disciplines other than classification (Smith-Miles, 2008). Our approach differs from metalearning in two ways. First, we capture expertise in ontologies and rule bases. Second, we make use of metadata in addition to properties computed from the data.

Another approach was taken by (Calcagno & de Mazancourt, 2010). In their paper, they describe an R package *glmulti* for the model selection problem with Generalized Linear Models (GZLM). *glmulti* focuses mainly on feature selection which is a sub-problem of automated model selection. *glmulti* takes the list of all predictor variables and generates all

possible unique models from the combination of these variables. After generating and fitting all models using a specified metric iteratively, *glmulti* returns top *n* models for the input. Since the number of possible unique models grows exponentially with respect to the number of variables in the input, their approach becomes prohibitive as fitting all models requires significant resources. The authors try to overcome this limitation by using a genetic algorithm to assist model generation and try not to fit all possible models. Finally, *glmulti* does not take into consideration metadata when reducing the sample space as opposed to our approach.

Traditional techniques for model selection include using decision trees that are hand crafted by domain experts[8]. These trees classify techniques based on the goal of the analysis and basic dataset properties such as number and types of predictors and responses. Based on the properties of the dataset to be analyzed, one can find a suitable technique (i.e., model type) for analysis. However, these approaches usually provide general outlines and leave the majority of the work including the implementation to the analyst. Even though it is possible to extend this approach, we suggest using an ontology-based approach. This approach has several advantages over a decision tree based approach. First, the expertise is captured in description logic. Capturing the expertise in a formal language such as description logic makes it easier to point out inconsistencies, validate suggestions and create executable models automatically. This may not carry much importance when the domain knowledge rarely changes, however, statistics is a vibrant field. Additionally, each scientific discipline has well-established preferences when it comes to statistical analysis. Therefore it is important to have an adaptable platform that is designed with changes in mind. Additionally, since model suggestion is done by a description logic reasoner, it is possible to generate formal justifications for why a particular model is suggested.

Finally, we look at the Assistant from Minitab® Statistical Software[9]. The Assistant is a tool available in the Minitab software that guides users through a step-by-step interactive workflow to help select the most appropriate model/technique for analyzing their dataset. In addition to helping select the most appropriate model, the Assistant provides visual representation of analysis results, explanation of different choices throughout the workflow and performs automated tests on the dataset and the selected model to ensure validity of results. The Assistant has many workflows for guiding one through hypothesis testing, design of experiments, regression and etc. However, support for predictive analytics is limited to multiple linear regression with up-to five predictor variables (only one of the predictors may be categorical) and a continuous response[10]. Additionally, only pair-wise interactions and quadratic expansions of the variables are considered during the model selection process.

[8]See (Tabachnick & Fidell, 2013) for an example. There are also websites that serve similar purpose. Examples include http://www.ats.ucla.edu/stat/mult_pkg/whatstat/ and http://www.socialresearchmethods.net/selstat/ssstart.htm.

[9]http://support.minitab.com/en-us/minitab/17/technical-papers/

[10]Refer to the following white paper for more information on how the model selection is performed. http://support.minitab.com/en-us/minitab/17/Assistant_Multiple_Regression.pdf

4 ScalaTion Framework

We use the ScalaTion (Miller et al., 2010) framework which supports multi-paradigm modeling for running our analyses. ScalaTion provides many analytics techniques for optimization, clustering, predicting, etc. which could be easily integrated in a large scale data analysis pipeline. Additionally, ScalaTion supports discrete event and continuous simulation.

ScalaTion is organized in three major package groups. The analytics package, `analytics`, includes implementations of major analytics algorithms which can be categorized under four types: predictors, classifiers, clusterers and reducers. Additionally, the `graphalytics` package provides implementations for graph-based analytics.

The optimization packages, `minima` and `maxima`, provide algorithms for optimization and implement major optimization paradigms such as Linear, Integer, Quadratic and Nonlinear Programming and the Simplex method. The minima package is for minimization, while the `maxima` package is for `maximization`.

Finally, the simulation packages provide simulation engines for a variety of different modeling paradigms. Currently, SCALATION has implementations for tableau, event, process, activity and state oriented models in addition to system dynamics. SCALATION also has 2D (and prototype 3D) visualization support for such models.

SCALATION, coded in the Java Virtual Machine (JVM) based the Scala language, makes use of Scala's native parallelism support via `.par` functions in addition to processing using Akka 11. In SCALATION, the `linalgebra.par` package, which currently contains 3,598 lines of code, contains parallel versions Cholesky and QR factorizations as well as parallel versions of many operations for both dense and sparse matrices. For some operations like matrix multiplication, the `.par` function available in Scala makes it easy to convert sequential to parallel code. Below is the definition of the matrix multiplication function in SCALATION's `MatrixD` class. The parallel ranges in the `for` loop split the iterations of the loop into manageable units of execution that can be performed by parallel threads.

```
def * (b: MatrixD): MatrixD = {
  val c = new MatrixD (dim1, b.dim2)
  val bt = b.t // transpose the b matrix
  for (i <- range1.par; j <- c.range2.par){
    val va = v(i); val vb = bt.v(j)
    var sum = 0.0
    for (k <- range2) sum += va(k) * vb(k)
    c.v(i)(j) = sum
  } // for
  c
} // end * function
```

For other operations such as Cholesky and QR factorizations, matrix inversion and SVD, a substantial speedup via parallelism is not so easy to achieve. For example, in (Lahabar & Narayanan, 2009) speedup of SVD was only achieved by utilizing Graphics Processing Units (GPUs) via the CUDA[12] platform. We plan on exploring the use of coprocessors (e.g., Intel Xeon Phi), custom thread pooling, and frameworks for distributed computation (e.g., Akka) to facilitate speedup.

There are also other modeling environments such as Weka (Hall et al., 2009) sharing similar functionality with SCALATION. Weka is a popular data analytics and machine learning platform written in Java. It provides a very intuitive user interface that allows pre-processing of data in addition to visualization. Weka also provides a Java API for programmatic access to the algorithms. In contrast to SCALATION, Weka is focused on machine learning. By providing an integrated approach, SCALATION reduces the cost of development time for a multi-paradigm modeling task.

In contrast to statistical software like R and SAS, SCALATION has a cleaner syntax that allows for mathematical formulas and expressions to more closely resemble their standard textbook notations. For example, Figure 2, Figure 3 & Figure 4 show how to estimate the coefficients in a Principal Component Regression (PCR) model of the form $\boldsymbol{y} = \boldsymbol{Xb} + \boldsymbol{\varepsilon}$ using SVD in R, SAS and SCALATION, respectively. Let U, Σ and V be the factors obtained from applying SVD to X. Then, an estimate for $\boldsymbol{b}$ is

$$\hat{\boldsymbol{b}} = V\Sigma^{-1}U^t y = V\frac{1}{\Sigma}U^t y$$

where Σ^{-1} and $\frac{1}{\Sigma}$ both represent the inverse of the diagonal matrix containing the singular values of X (Mandel, 1982). Of the three code examples provided, one could argue that the SCALATION example is not only more readable, but looks closer to the mathematical notation provided above.

Additionally, since SCALATION is written in Scala for the JVM, users can easily integrate existing APIs in Scala, Java, and other JVM languages. For integration with native code libraries, the Java Native Interface (JNI) can be used. While Spark is also available on the JVM, it utilizes Stochastic Gradient Descent (SGD) to produce least squares estimates in regression whereas SCALATION utilizes factorization techniques such as Cholesky Factorization, QR Decomposition and SVD Decomposition which are commonly considered to work better in practice.

5 Extraction of Metadata

In order to reduce model type space and find suitable modeling techniques for a given dataset, some information needs to be gathered. Some of this information can be obtained directly by performing a quick analysis of the dataset. We list a number of properties in Table I that can be obtained in such a way.

[12]CUDA: http://www.nvidia.com/object/cuda_home_new.html

The domains of variables are essential when choosing an appropriate model type. For example, if the domain of the response variable is binary, then a logistic regression model can be selected. Similarly, for a dataset which contains both discrete and continuous predictor variables, an ANCOVA model can be more appropriate. Although capturing the domain of a variable can be difficult, it can be useful when this information is not available in the metadata. For example, a technique based on repeated differences can be employed to help identify whether the values are discrete.

Diagnostic analysis of variables includes computing descriptive statistics such as mean and variance as well as more advanced information such as the probability distribution of the variable. This information may be useful in several ways. First, it can help the model suggestion process. As an example, *Poisson Regression* may be used for modeling count data (i.e., non-negative integer response).

However, an important assumption for Poisson regression is that the mean is equal to the variance. If the diagnostics reveal that the variance is significantly larger than mean (i.e., overdispersion) negative-binomial regression may be suggested instead. It may also help determine the validity of a selected model after fitting.

A very important factor in predictive analytics is reliability. (Harrell, 2015) defines reliability as the ability of the fitted model to predict future instances as well as existing instances upon which the model is trained. If a model is overfitted, its reliability of predicting future instances will be low. In addition to choosing an inappropriate model, a major cause for overfitting is having too many predictors for the given sample size. According to (Harrell, 2015), a fitted relation model is likely to be reliable in most cases if the number of predictors p is less than $m/15$ where m is the limiting sample size. m is equal to the number of samples in the dataset for a continuous response. For a binary response, m is equal to min (n_1, n_2) where n_1 and n_2 are marginal frequencies of two response levels. In cases where $p > m/15$, a dimension reduction approach such as Principal Component Analysis (PCA) may be taken. PCA can also be performed to reduce the dimensionality of the matrix when the data matrix for the predictor variables has high *multicollinearity*. Additionally, this step can help make some of the underlying matrix operations performed by various algorithms easier (e.g., matrix factorization). Another consideration that should be taken with respect to the data matrix is whether or not it is sparse. Certain algorithms can take advantage of specialized data structures for storing sparse matrices in order to reduce memory consumption and avoid unnecessary operations (Smailbegovic, Gaydadjiev, & Vassiliadis, 2005).

Additionally, any available metadata can also be used. As mentioned in the related work, existing systems operate solely on the input data ignoring problem definition, field descriptions and other metadata. However, predictive analytics often requires knowledge of the experiment design which was used in order to generate the data. Additionally, selection of a modeling technique is often dependent on the goal of the analysis.

Table II lists a number of useful properties that can potentially be obtained from metadata. An associated task can be as simple as whether this problem is a prediction or a

classification problem. In some cases, the Associated Task could be as specific as *ANOVA* or *ANCOVA* (e.g., if you are interested in the relationship between the predictors themselves with respect to the response). Often times, data is stored in a matrix format where a row indicates a sample and columns indicate the features of the sample. The Response Column indicates the column index of the data matrix in which the response variable is stored.

Variable Type indicates the domain of a variable (e.g., continuous, binary, ordinal, etc.) and can be extracted for each variable in the dataset depending of the availability of the metadata. Whether the dataset contains missing values or not and which variables have missing values can affect the selection of appropriate model types. Depending on the situation, our system may choose a model that can handle missing values or otherwise samples containing missing values would be discarded.

Whether a dataset has multivariate response or predictors can be implicitly available given the above information, however, the provided metadata may contain only partial information.

6 Analytics Ontology

The Analytics Ontology[13] is an ontology for supporting automated model selection. Currently, it is more geared towards predictive analytics. However, the ontology is designed to be extensible in order to support a wide range of analytics paradigms including classification and clustering.

The most important class in the ontology is *Model*. The *Model* class along with its sub-classes is both indicative of a model type as well as a partial realization of a statistical model. This is due to the fact that all other classes such as *Variable*, *Function*, and *Distribution* describe a part of the model.

Figure 6 displays the major classes and their hierarchy. We now give working definitions of the major classes of the ontology in the analytics context.

Model defines different model types that can be used for analyzing data. There are many ways of specifying the class hierarchy for a collection of model types, however, we have given priority to correspondence with implementations of these types (e.g., SCALATION). This becomes important when running models generated from the abstract models represented in the ontology. There are two top level Model classes, namely *DependentModel* and *IndependentModel*.

The main distinction between the two models is the dependency (i.e., correlation) among responses of the observations belonging to the same individual in the dataset. The dependency usually occurs when there are repeated observations (i.e., measurements) of the same individual in time or space dimension. Time-series models are a typical example of a *DependentModel*. Other major members of *DependentModel* are Generalized Estimating Equation (GEE) models and Generalized Linear Mixed Models (GLMM).

[13]Analytics Ontology can be accessed from https://github.com/scalation/analytics.

IndependentModel currently only includes Generalized Linear Models (GZLM) (Nelder & Baker, 2004). The most basic independent model is *Simple Linear Regression*, which quantifies the linear relationship between the response and a single explanatory variable. A similar modeling technique to regression is ANOVA, which targets the problem from a different angle that focuses on analyzing the differences among group means and their associated procedures. The extension of simple linear regression and ANOVA are multiple linear regression and multiple-factor ANOVA, respectively, which are used when there are two or more predictors. Those models, together with other models, such as before-mentioned ANCOVA, and polynomial regression that describe the linear relation between predictors and responses, belong to the general linear model class. The generalized linear model is a flexible generalization of general linear model that allows for response variables that have residual distributions other than normal distributions. The common generalized linear models include *Logistic Regression*, *Poisson Regression*, *Log-Linear Models*, etc. Changing the relationship between the parameters and the linear predictor, i.e., changing the link function, usually will lead to different generalized linear models.

Variable represents a feature of a *Model*. A variable can have a role of a predictor or a response which is defined by the relationship with *Model*. This relationship is defined by the object properties *hasPredictorVariable* and *hasResponseVariable*. Additionally, a predictor variable can also have a role of representing the time component in a time series model. In that case, *hasTemporalVariable* property may be used.

Variable Type restricts a *Variable*'s corresponding domain of discourse. The relationship between a *Variable* and *Variable Type* is defined by the object property *hasVariableType*. As seen in Figure 6, a *Variable* can either have a *Quantitative* or *Qualitative* type (Somun-Kapetanović, 2011). *Quantitative* types are *Continuous* and *Discrete*. Similarly, subclasses of *Qualitative* are *Nominal* and *Ordinal*. Finally, *Dichotomous* is a subclass of *Nominal*.

In some families of models such as GZLM (Generalized Linear Models) a link *Function* is used to relate the predictor variables to the mean response. As an example, *logit* is the most commonly used link function for *Logistic Regression*, whereas *log* is commonly used for *Poisson Regression*. The relationship between a *Model* and a *Function* is defined by the object property *hasLinkFunction*.

Distribution class refers to the (probability) distribution of a random variable which specifies the probability of occurrence for each outcome in the population represented by the random variable. Considering the fact that the residual (error term) of a model is considered to be a random variable, this class is used to specify the residual distribution of a model. This relationship can be defined using the *hasResidualDistribution* object property.

6.1 Equivalence Class Axioms

In addition to the class axioms (i.e., class definition, hierarchy of classes) and object properties, the Analytics Ontology defines equivalence class axioms to capture key characteristics of different model types. According to the OWL language specification, an equivalent class axiom states the equivalence of two named classes which is defined with

Description Logic syntax. Using these axioms, it becomes possible to deduce implicit information hidden in the ontology with a reasoner.

Figure 7 lists a few of the equivalence class axioms from the ontology. One can notice that *IndependentModel* and *GZLM* share the same axiom. This is due to *GZLM* being only *IndepedentModel* in the ontology currently.

Based on these axioms it is possible to infer that a dataset can be modeled using *ANOVA* if it has only *qualitative* predictor variables and a continuous response variable. We can easily see that *Multiple Linear Regression* is left out according to the equivalence axiom restricting the domain of at least one predictor variable to be *Continuous*. We should note that the restrictions defined by the equivalence axioms are used for elimination of unsuitable models. Therefore, a dataset can be modeled by using any other *Model* which is not eliminated by the reasoner during inference.

6.2 Model Selection Using Semantic Reasoning

We use description logic reasoning for reducing the model type space and obtaining suggestions. The process of reasoning with description logic may be defined as follows. Given background knowledge and observations, an explanation is computed. In our context, background knowledge is realized with the axioms in the ontology. These include the class axioms which define the hierarchy between model types and the equivalence class axioms. Similarly, observations are the acquired characteristics of the dataset of focus. These are captured by the facts which are defined by linking the instances describing the dataset to the ontology axioms using the object property relationships. Finally, explanation constitutes the set of *inferred* possible model types that could be used for analyzing the specified dataset.

Our reasoning framework is fully captured in the Analytics Ontology which is expressed in OWL 2 DL profile. The DL profile imposes certain restrictions over the full OWL 2 language so that certain computational guarantees can be made by the reasoners which support reasoning with OWL 2 DL profile. Since the Analytics Ontology fully adheres to OWL 2 DL profile, any existing OWL 2 reasoner (e.g., HermiT (Shearer, Motik, & Horrocks, 2008), Pellet (Sirin, Parsia, Grau, Kalyanpur, & Katz, 2007), etc.) can be used for deducing the model types for a given dataset. Also, this ensures that our approach is decidable as all possible inferences are guaranteed to be made. A logical reasoner serves several different purposes for an ontology. It performs consistency checking for all axioms in the ontology including the axioms that were inferred during the discovery phase. This step help ensure that the domain expertise is properly captured in the ontology and may be used for suggesting suitable models for datasets.

Additionally, it helps extend the knowledge base by discovering hidden (i.e., implicit) information related to the concepts and individuals in the ontology. As an example, given the axiom "*mpg* hasVariableType *Non-Negative Continuous*", it is possible to infer the following axiom "*mpg* hasVariableType *Continuous*" even though it was not originally included in the ontology. In the case of a dataset, the information of the most suitable model type(s) is hidden in the characteristics and properties of the dataset that are expressed in the

ontology. These *hidden* (implicit) information is discovered by the logical reasoner from the explicit information added to the ontology.

6.3 ScalaDash

In order to demonstrate the approach described in this paper, we have developed SCALADASH[14]. Developed in the Scala language in order to leverage the SCALATION framework to full extent, SCALADASH is an application with a graphical user interface that is designed to handle all steps of the predictive analytics workflow as described in Section 2.

We use the Auto-MPG dataset (Quinlan, 1993) from UCI Machine Learning Repository (Lichman, 2013) as a small example for illustration. The dataset is for prediction of fuel consumption in miles per gallon and contains 3 discrete variables; cylinders, model year, origin and 5 continuous variables; mpg, displacement, horsepower, weight and acceleration. The dataset contains 398 samples including 6 samples with missing values for horsepower. According to this specification, the dataset can be represented in the ontology as shown in Figure 8. By default, all individuals are an instance of *owl:Thing* in OWL. Note that the *AutoMPGModel* is not an instance of any specific *Model* class in the hierarchy as this is not explicitly expressed (see Figure 8). It is not known at this point which models are suitable for this dataset.

As opposed to other logic frameworks OWL takes an *open-world assumption* (OWA) when dealing with missing/incomplete knowledge. In a *closed-world assumption* (CWA) system, any information that does not exist in the knowledge is considered false[15]. As an example, in a *closed-world*, a logical reasoner can deduce that *AutoMPGModel* only has 8 variables as shown in Figure 8. However, an open-world reasoner cannot deduce the same fact as there might be other variables which may still exist in another knowledge base. For this reason, besides asserting the facts as in Figure 8, we also need to assert closure axioms that are implicitly asserted in a *closed-world assumption* (CWA). Following the same example, the following assertions must be added to the ontology to “close” the axioms for *AutoMPGModel*.

- *AutoMPGModel* hasVariable only *({acceleration, cylinders, displacement, horsepower, model_year, mpg, origin, weight})*
- *mpg* hasDistribution *only ({Normal_Distribution_Instance})*
- *mpg* hasVariableType only *({Non_Negative_Continuous_Variable_Type})*[16]

Another related characteristic of OWL is the lack of *unique name assumption* (UNA). As a result, OWL makes no assumption that individuals with different unique names are in fact different individuals. To let the reasoner be aware of the fact that the variables are different, we add the following axiom to the ontology.

- DifferentIndividuals ({*acceleration, cylinders, displacement, horsepower, model_year, mpg, origin, weight*})

[14] SCALADASH is available for download at https://github.com/scalation/analytics
[15] For a more in-depth discussion on OWA and CWA, see (Russell, Norvig, Canny, & Bratko, 2005)
[16] Similar assertions are made for other variables as well. They are excluded for brevity.

After the *AutoMPGModel* instance is properly closed, the following facts about the dataset are inferred when a reasoner is run on the ontology:

- *AutoMPGModel* is-a *Model*.
- *AutoMPGModel* is-a *IndependentModel*.
- *AutoMPGModel* is-a *GZLM*.
- *AutoMPGModel* is-a *GLM*.
- *AutoMPGModel* is-a *ANCOVA*.
- *AutoMPGModel* is-a *Multiple Linear Regression*.

As one can quickly notice, some of these inferences such as "*AutoMPGModel* is-a *Model"* do not carry much importance in terms of information content. Therefore, a filtering based on the model hierarchy is performed. An inference is not added to the list of suggestions if one of its subclasses already exist in the list. The pseudocode for the filtering algorithm is given in Figure 9. After the filtering function is run, only the following inferences are returned to the user as suggestions as shown in Figure 10:

- *AutoMPGModel* is-a *ANCOVA*.
- *AutoMPGModel* is-a *Multiple Linear Regression*.

Based on the current knowledge, all the inferred models can be suitable for analyzing this dataset. However, *additional information can be asserted as new information* becomes available. For example, during the full model residual analysis phase after the candidate models were run, if the residual distribution of the model is determined to be an exponential distribution we add this information by asserting the following statement "*AutoMPGModel* hasResidualDistribution *Exponential Distribution*" to the ontology. When the reasoner is re-run including this new information, *Exponential Regression* is added to the list. Also note that *ANCOVA* and *Multiple Linear Regression* are no longer possible models since both models are a sub-class of *GLM* model which expects the residuals to be *Normally Distributed*.

Based on the suitable models inferred by the reasoner, running a model with SCALATION can be performed in the following fashion. Given a data matrix X and response vector $\boldsymbol{y}$, a Multiple Linear Regression model can be run as shown in the code snippet below.

```
val mpgMLRModel = new Regression (x, y)
mpgMLRModel.train ()
println ("fit = " + mpgMLRModel.fit)
```

The `fit` function returns coefficient estimates and as well as various model diagnostics (e.g., R^2 and $F-Statistic$, etc.). The R^2 fit for the full *AutoMPG* model (including all predictors) is 0.8093. After performing *Backwards Elimination* for basic feature selection with a 5% drop in R^2 as the stopping threshold, the final reduced model gives an R^2 *of* 0.8082. The final model only contains variables *weight* and *model_year*.

In a different example, we look into the resin defects dataset[17]. The dataset captures three predictor variables: *hours* since last hose cleaning, *temperature* of the resin and the *size* of the screw that moves the resin pellets through the hoses, to predict the number of *discoloration* defects per hour during the manufacturing process. Since the response variable represents a count (Non-Negative Integer), *Poisson Regression* is inferred as the suitable model by our system. In contrast, *glmulti* tool does not have a mechanism to capture information of this nature and therefore suggests a general linear model (glm) formula. A quick comparison of full models using R^2 reveals that *Poisson Regression* model provides a better fit for this dataset.

6.4 Incorporation of knowledge through domain ontologies

Since the Analytics Ontology is written in the OWL language, it can be easily supplemented with a domain ontology related to the input domain. Besides top-level ontologies such as Dublin Core (DC) that define common metadata terms, many domain-specific ontologies have been created and been in use extensively. It is becoming more common that data is published with metadata defined in these domain-specific ontologies especially in the scientific and biomedical domains. As a prominent example, the Gene Ontology (GO) contains over 40,000 biological concepts and has been used for annotating more than 100,000 peer-reviewed scientific papers with information on genes and gene properties[18]. The well-defined knowledge captured in these domain ontologies would assist reducing the possible model space even further. For example in a traffic dataset, the response variable might capture a *count* such as number of cars passing by an intersection in a certain period of time. Even though the count is expected to be *Non-Negative Integer*, there might be cases where it is represented in a different domain such as *Continuous*. For example, the traffic count might be captured from multiple sensors and the aggregation of sensory data may result in having fractional values. In such situation, even though the domain of the variable is not *Non-Negative Integer*, it can still be modeled using *Poisson Regression* since the response variable captures a *count*. Many insights such as in the example are available in domain ontologies. Additionally, it can also be beneficial for interpreting the final model since an alignment provides more information such as the how the variables are related and etc.

7 Conclusion

We have described a framework for supporting semi-automated model selection and model execution for conducting predictive analytics. Particularly, we show that instance classification can be used for reducing the set of applicable model types. After the model types are chosen, feature selection can be performed in order to find suitable subsets of the full model for each type. This is important because as the number of predictor variables and/or the sample size increase, exploration of all possible model types becomes less feasible. The ability to provide an explanation of why a particular model is selected based upon the inference provided by the reasoner is an important advantage over existing model

[17] http://support.minitab.com/en-us/datasets/
[18] Gene Ontology: http://geneontology.org/page/about

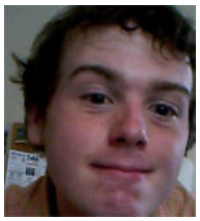
selection tools. For future work, we plan to extend the ontology with concepts (e.g., skewness, kurtosis, etc.) to assist automating workflow steps after model selection. Additionally, we plan to facilitate alignments with top-level and domain-specific ontologies to aid interpretation of analysis results. Finally, we plan to conduct extensive evaluations of the usability of this approach and provide performance results.

Acknowledgments

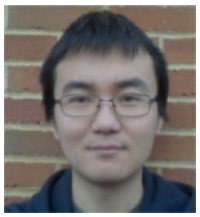
PM and RX were partially supported by NSF grants DMS-1440037, 1438957, 1440038 and NIH grant 1R01GM113242-01. PM currently is supervising the research of Ph.D. student RX.

Biographies

Mustafa V. Nural is a Ph.D. candidate in the Department of Computer Science at The University of Georgia. His research interests include semantic web and big data analytics. Nural received the B.S degree in Computer Science from Fatih University in Turkey in 2008. Nural is currently working as a Data Developer in NIH-funded multi-institute MaHPIC (Malaria Host-Pathogen Interaction Center) project led by Emory University.



Michael E. Cotterell is an Instructor and Ph.D. student in the Department of Computer Science at The University of Georgia. He received his B.S. in Computer Science from UGA in May 2011. As an undergraduate, he served as Vice Chairman of the UGA Chapter of ACM. In 2012, he interned at the National Renewable Energy Lab (NREL), where he conducted research on formal ontologies for Energy Systems Integration and the Energy Informatics domain. In 2014, he was a recipient of the NSA Mathematics and Computer Science Student Scholarship. His research interests include Simulation, Big Data Analytics, and Formal Ontologies with inter-disciplinary applications related to Statistics and Informatics.



Hao Peng is a Ph.D. student in the Department of Computer Science at The University of Georgia. His research interests are in the field of big data analytics, and in particular, parallel learning of probabilistic graphical models. Hao received the B.S. degree in both Computer Science and Statistics Summa Cum Laude and with Highest Honors from the University of Georgia in 2013.

Rui Xie is a Ph.D. candidate in the Department of Statistics at The University of Georgia. His research interests include time series and sequential statistical analysis, bioinformatics and big data analytics.

Ping Ma is a Professor in the Department of Statistics at The University of Georgia. His research interests include Bioinformatics, Functional Data Analysis, and Geophysics. Dr. Ma received the Ph.D. degree in Statistics from Purdue University in 2003.

John A. Miller is a Professor of Computer Science at the University of Georgia. He has been the Graduate Coordinator for the department for 9 years and is currently the Associate Head. His research interests include (i) Modeling & Simulation, (ii) Web Services/Workflow, (iii) Database Systems, and (iv) Big Data Analytics. Dr. Miller received a B.S. in Applied Mathematics from Northwestern University in 1980 and an M.S. and Ph.D. in Information

and Computer Science from the Georgia Institute of Technology in 1982 and 1986, respectively. As part of his co-operative undergraduate education program, he worked as a Software Developer at the Princeton Plasma Physics Laboratory. In his areas of interest, Dr. Miller has authored of over 180 research papers. He is an Associate Editor for the ACM Transactions on Modeling and Computer Simulation, SIMULATION: Transactions of the Society for Modeling and Simulation International, and previously IEEE Transactions on Systems, Man and Cybernetics as well as an Editorial Board Member for the Journal of Simulation and International Journal of Simulation and Process Modelling.

References

Bernstein, A., Hill, S., Provost, F. Intelligent Assistance for the Data Mining Process: An Ontology-based Approach. New York, NY, USA: 2002. CeDER Working Paper IS-02-02

Calcagno V, de Mazancourt C. glmulti: An R Package for Easy Automated Model Selection with (Generalized) Linear Models. Journal of Statistical Software. 2010; 34(12):1–29.

Codd EF. A Relational Model of Data for Large Shared Data Banks. Commun ACM. 1970; 13(6):377–387. http://doi.org/10.1145/362384.362685.

Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. Ann Statist. 2004; 32(2):407–499. http://doi.org/10.1214/009053604000000067.

Godambe, VP. Estimating Functions. New York: Oxford University Press; 1991.

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA Data Mining Software: An Update. SIGKDD Explor Newsl. 2009; 11(1):10–18. http://doi.org/10.1145/1656274.1656278.

Harrell, FEJ. Regression Modeling Strategies. New York, NY: Springer New York; 2015. Multivariable Modeling Strategies; p. 67-75.http://doi.org/10.1007/978-1-4757-3462-1_4

Lahabar, S., Narayanan, PJ. Singular value decomposition on GPU using CUDA; Parallel Distributed Processing, 2009 IPDPS 2009 IEEE International Symposium on. 2009. p. 1-10.http://doi.org/10.1109/IPDPS.2009.5161058

Lichman, M. {UCI} Machine Learning Repository. 2013. Retrieved from http://archive.ics.uci.edu/ml

Mandel J. Use of the Singular Value Decomposition in Regression Analysis. The American Statistician. 1982; 36(1):15–24.

Miller JA, Han J, Hybinette M. Using domain specific language for modeling and simulation: scalation as a case study. Proceedings of the Winter Simulation Conference. 2010:741–752. Winter Simulation Conference.

Nelder, JA., Baker, RJ. Encyclopedia of Statistical Sciences. John Wiley & Sons, Inc.; 2004. Generalized Linear Models.

Quinlan, JR. Machine Learning Proceedings of the Tenth International Conference. Morgan Kaufmann; 1993. Combining instance-based and model-based learning; p. 236-243.

Reif M, Shafait F, Goldstein M, Breuel T, Dengel A. Automatic classifier selection for non-experts. Pattern Analysis and Applications. 2012; 17(1):83–96. http://doi.org/10.1007/s10044-012-0280-z.

Russell, SJ., Norvig, P., Canny, J., Bratko, I. Artificial Intelligence: A Modern Approach. Pearson Education, Limited; 2005.

Shearer, R., Motik, B., Horrocks, I. HermiT: A Highly-Efficient OWL Reasoner. In: Dolbear, C.Ruttenberg, A., Sattler, U., editors. Proceedings of the Fifth OWLED Workshop on OWL: Experiences and Directions, collocated with the 7th International Semantic Web Conference (ISWC-2008); Karlsruhe, Germany. October 26–27, 2008; 2008. CEUR-WS.org

Sirin E, Parsia B, Grau BC, Kalyanpur A, Katz Y. Pellet: A practical OWL-DL reasoner. Web Semantics: Science, Services and Agents on the World Wide Web. 2007; 5(2):51–53. http://doi.org/10.1016/j.websem.2007.03.004.

Smailbegovic FS, Gaydadjiev GN, Vassiliadis S. Sparse matrix storage format. Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing, ProRisc 2005. 2005:445–448.

Smith-Miles KA. Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Computing Surveys. 2008; 41(1):1–25. http://doi.org/10.1145/1456650.1456656.

Somun-Kapetanović, R. Variables. In: Lovric, M., editor. International Encyclopedia of Statistical Science. Springer Berlin Heidelberg; 2011. p. 1639-1640.http://doi.org/10.1007/978-3-642-04898-2_99

Stonebraker M, Abadi DJ, Batkin A, Chen X, Cherniack M, Ferreira M, Zdonik S. C-store: a column-oriented DBMS. 2005:553–564.

Tabachnick, BG., Fidell, LS. Using Multivariate Statistics. 6th. Needham Heights, MA, USA: Allyn & Bacon, Inc; 2013.

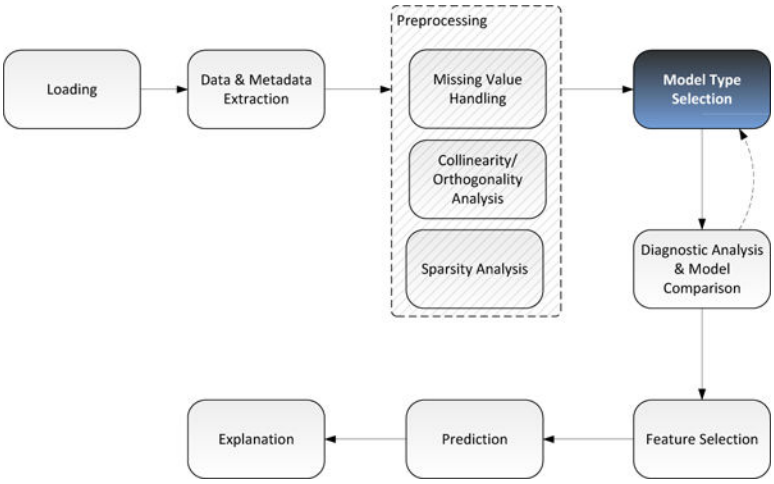


Figure 1.
Predictive Analytics Workflow.

```
svd <- svd(x)
u <- svd$u
d <- diag(svd$d)
v <- svd$v
b <- v %*% qinv(d) %*% t(u) %*% v
```

Figure 2.
Estimating Coefficients of a PCR Model Using SVD in R.

```
call svd(u, d, v, x);
b = v * inv(diag(D)) * u` * y;
```

Figure 3.
Estimating Coefficients of a PCR Model Using SVD in SAS.

```
val (u, d, v) = x.svd.factors()
val b = v ** d.recip * u.t * y
```

Figure 4.
Estimating Coefficients of a PCR Model Using SVD in ScalaTion.

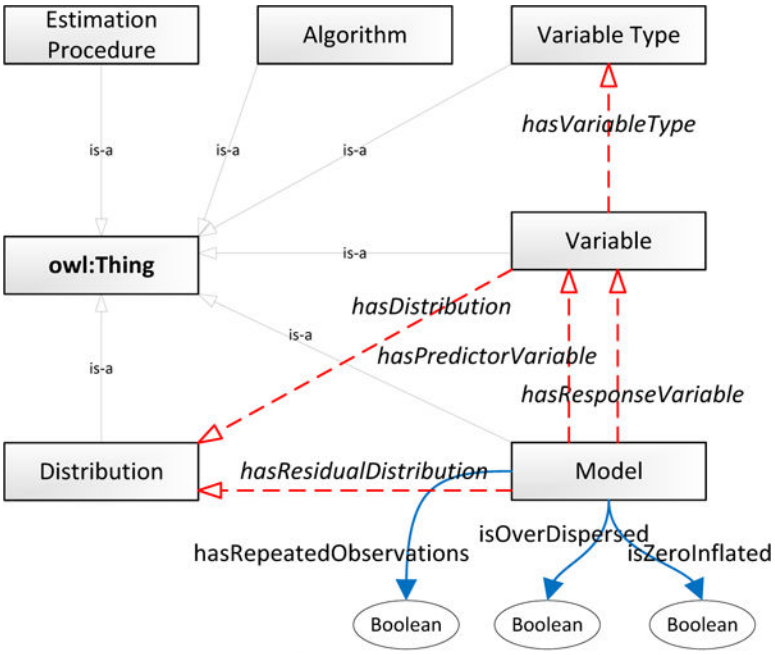


Figure 5.
Main Object & DataType properties in the analytics ontology

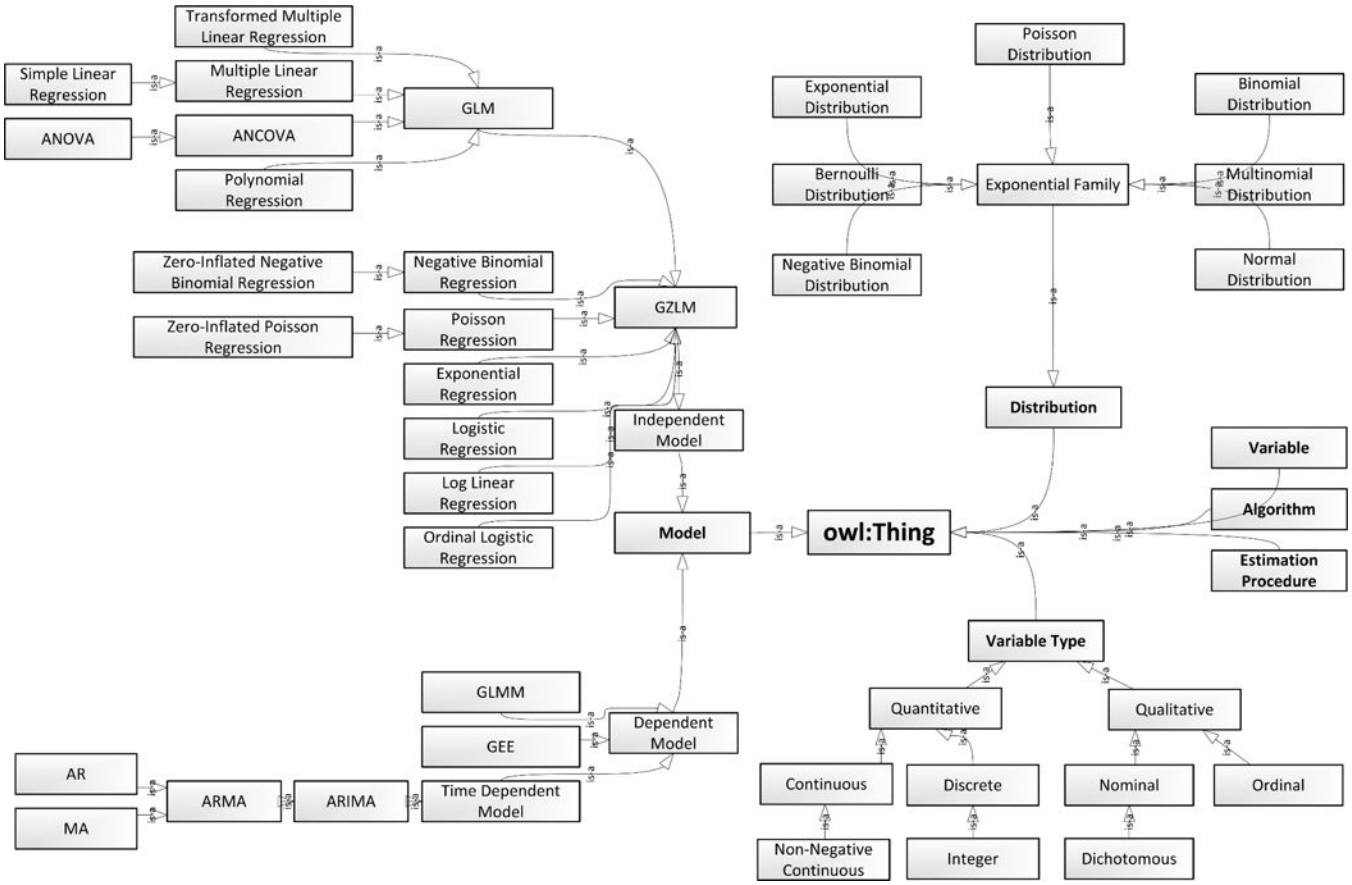


Figure 6.
Partial View of the Analytics Ontology (Only Shows Class Hierarchy)

```
IndependentModel ≡
        Model and (hasRepeatedObservations value false)
GZLM ≡
        Model and (hasRepeatedObservations value false)
Poisson Regression ≡
        GZLM
        and (
                (hasResidualDistribution only 'Negative Binomial Distribution') or
                (hasResidualDistribution only 'Poisson Distribution') or
                (hasResidualDistribution exactly 0 Thing)
        )
        and (hasPredictorVariable some (hasVariableType only Continuous))
        and (hasResponseVariable only (
                (hasDistribution only 'Poisson Distribution')
               and (hasVariableType only 'Non-Negative Integer')
        ))
Exponential Regression ≡
        GZLM
        and ((hasResidualDistribution only 'Exponential Distribution') or
        (hasResidualDistribution exactly 0 Thing)
        )
        and (hasPredictorVariable only (hasVariableType only Continuous))
        and (hasResponseVariable only (hasVariableType only 'Non-Negative Continuous'))
GLM ≡
        GZLM
        and (
                (hasResidualDistribution only 'Normal Distribution')
                 or (hasResidualDistribution exactly 0 Thing)
       )
        and (hasResponseVariable only (hasVariableType only Continuous))
Multiple Linear Regression ≡
        GLM
        and (hasPredictorVariable some (hasVariableType only Continuous))
        and (hasResponseVariable only (hasDistribution only 'Normal Distribution'))
ANCOVA ≡
        GLM
        and (hasPredictorVariable some (hasVariableType only Qualitative))
        and (hasResponseVariable only (hasDistribution only 'Normal Distribution'))
```

Figure 7.
Some of the Equivalence Class Axioms from the Ontology

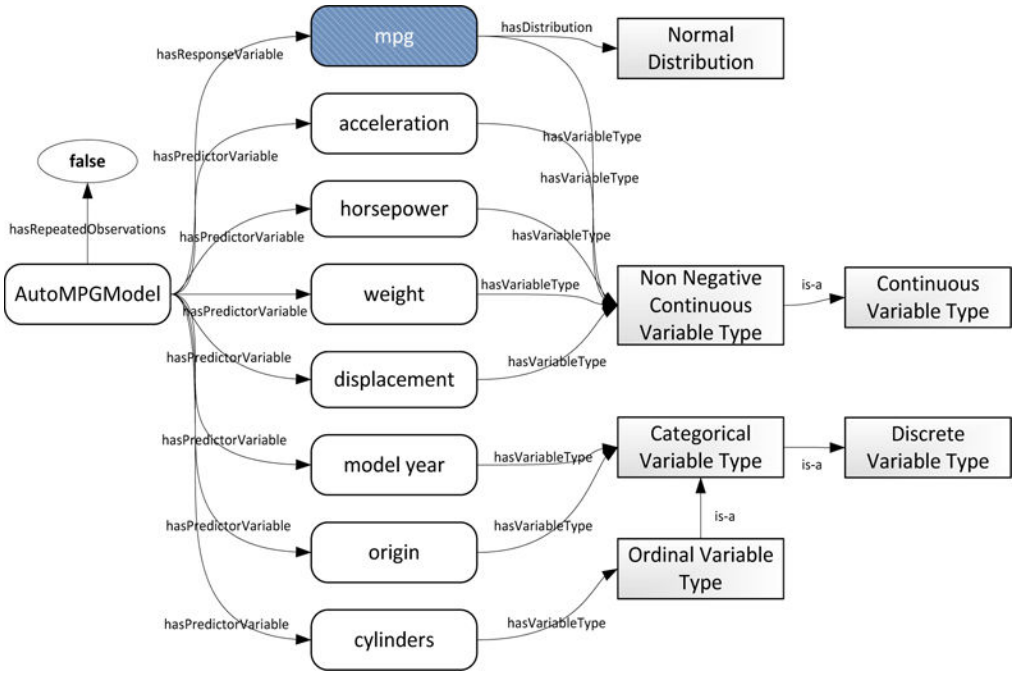


Figure 8.
Representation of AutoMPGModel in the Ontology

```
function FILTER (datasetId, ontology)
 suggestions = retrieveTypes(datasetId, ontology)
 for s in suggestions
  subs = getSubclasses(s, ontology)
  for sub in subs
   if sub in suggestions
     removeFromSuggestions
   end if
  end for
 end for
 return suggestions
```

Figure 9.
Algorithm for Filtering Suggestions

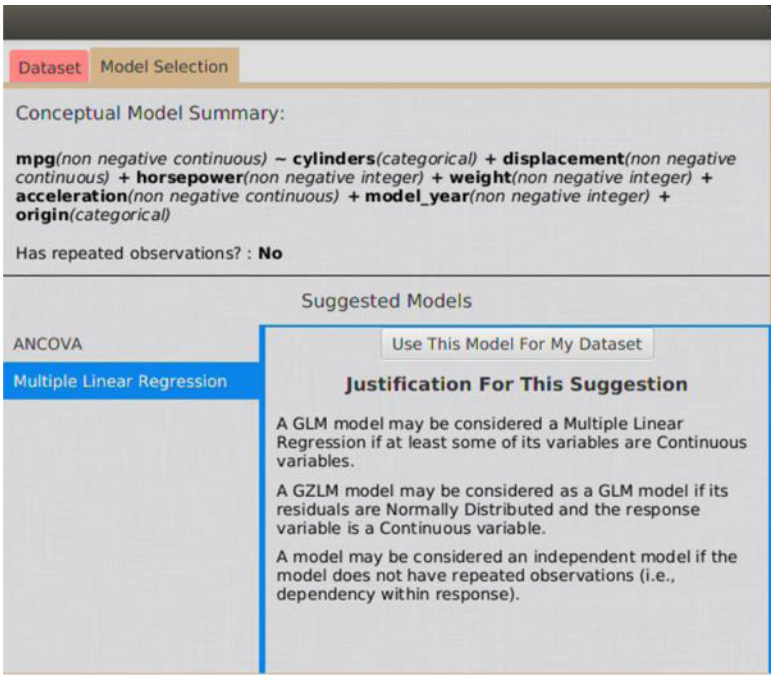


Figure 10.
A Screenshot from SCALADASH Displaying Suggestions for AutoMPGModel

Table I

Data Extraction Tasks

TASK	DESCRIPTION
Domain of Variables	Continuous, Discrete, Binary, etc.
Variable Diagnostics	Mean, variance, probability distribution and etc.
Dimension Analysis	Dimension reduction based on multicollinearity/rank (PCA etc.)
Feature Space Analysis	Analyzing relationship between sample size and number of predictors to prevent overfitting
Sparsity	Yes, No
# of Samples	Integer

Table II

Metadata Extraction Tasks

TASK	DESCRIPTION
Associated Task (Goal)	Prediction, Classification, etc.
Multivariate Response?	Yes, No
Multivariate Predictor?	Yes, No
Variable Type	Continuous, Discrete, Binary, etc.
Response Column	Column number(s) of the response variable
Missing Values	Yes, No