A List of Metrics from the expose2 Ontology (148 Classes)

`http://expdb.cs.kuleuven.be/Expose.owl`

By: Noureddin Sadawi

10 Feb 2014

1) **DM_entity_property**
    A) **algorithm_property**
        I) **qualitative_algorithm_property**
            1) comprehensibility: Is it easy for users to understand how the algorithm works and what it parameters do?
            2) cumulativity: Algorithms which do not allow attributes to interact, such as naive Bayes have low cumulativity, while decision trees have high cumulativity, allowing maximal interaction between variables
            3) **feature_handling**
                a) mixed_feature_handling
                b) parallel_feature_handling
                c) sequential_feature_handling
            4) handles_cost_functions
            5) handles_nominal_features
            6) handles_nominal_target
            7) handles_numeric_target
            8) handles_numerical_features
            9) handles_relational_data
            10) incrementality: An algorithm is incremental if it can build and refine a model gradually as new training instances come in, without reexamining all instances seen in the past. However, they are generally very sensitive to the order in which examples are presented.
            11) interpretability: Whether the model can be interpreted to extract knowledge. For instance, a decision tree is easy to interpret by humans, a neural network is not.
            12) **model_refinement_procedure**

a) hypothesis_optimization: An hypothesis optimization learner is an inductive algorithm that assumes the data can be correctly modelled by a given mathematical function, and that adjusts that function (e.g. by adjusting weights) to fit the given data optimally. Examples are linear regression, backpropagation in artificial neural networks, Bayes rule in Bayesian methods and the kernel trick in kernel methods.

b) hypothesis_refinement: An hypothesis refinement learner is an inductive algorithm that assumes the data can be correctly modelled by a given model, and that incrementally adjusts that model as it sees new data points (observations). Examples are decision tree learners, covering learners (rule learners), kMeans and logical induction.

13) optimizability: The degree to which model and search parameters are handled automatically, or how much work goes into fine-tuning all parameter settings. Naive Bayes is very simple to tune, Support Vector Machines are much harder to optimize

14) performs_classification

15) performs_regression

16) representational_model

II) **quantitative_algorithm_property**

1) bias-variance_profile: Bias and variance can be measured only with regard to the expectation of a learned models actual output, averaged over the ensemble of possible training sets D for a fixed sample size N. We thus need to evaluate algorithms on many different equally-sized samples of the same dataset, and repeat this process over many datasets in order to assess the average percentage of bias error versus variance error typically observed in the algorithm. If an algorithm has parameters that influence its bias, then we must take these into account as well.

2) resilience_to_irrelevant_attributes

3) resilience_to_missing_values

4) resilience_to_noise

5) resilience_to_redundant_attributes

6) scalability: A learning algorithm is scalable if it performs as well on large datasets as it does on smaller ones. A systematic profiling should include the dimensions of the dataset, such as the number of examples, the number of attributes, and in case of classification, the number of classes.

7) space_complexity
8) time_complexity

B) **data_property**

I) **dataset_property**

1) **qualitative_dataset_property**

a) **data_cleanliness**

i) contains_missing_values
ii) missing_value_free

b) **supervision**

i) semi-supervised
ii) supervised
iii) unsupervised

2) **quantitative_dataset_property**

a) **concept-based_dataset_property**

i) concept_variation: the (non-) uniformity of the class-label distribution throughout the feature space (measured through the distance between two examples of a different class)
ii) example_cohesiveness: the density of the example distribution in the training set

b) **information_theoretic_dataset_property**

i) average_mutual_information
ii) coefficient_of_variation_of_target: The coefficient of variation of the target is defined as the ratio of the standard deviation to the mean of the target attribute and can be used instead of entropy on numerical targets. It is a normalization of the standard deviation of the target useful for numerical targets X_{target}. A related measure, *sparsity of the target*, is $VarCoef_{target}$ discretized into 3 values.

$$VarCoef_{target} = \frac{\sigma_X}{\mu_X} \tag{1}$$

iii) equivalent_number_of_attributes: The equivalent number of attributes is a quick estimate of the number of attributes required, on average, to describe the class.

$$EN - atrr = \frac{H(C)}{MI(C, X)} \tag{2}$$

iv) median_of_uncertainty_coefficients

v) noise_to_signal_ratio: The noise to signal ratio is an estimate of the amount of non-useful information in the attributes regarding the class. $\overline{H(X)}$ is the average information (useful or not) of the attributes.

$$NS - ratio = \frac{\overline{H(X)} - \overline{MI(C,X)}}{\overline{MI(C,X)}} \quad (3)$$

vi) target_feature_entropy

c) **instance-based_property:** Compares entire instances with each other. A dataset may contain two observations with similar or equal attribute values, but with different labels which might cause a classifier to get confused. Analogously, there might be two or more observations which are identical, which (unfairly?) gives them more weight in some algorithms. Here, properties derived from case-based learning are used originally intended to assess the quality of a given case-base.

i) instance_consistency: A single example is consistent within the dataset if and only if there does not exist any other example that is identical, but has a different target value

ii) instance_incoherence: Incoherence is a measure for how dissimilar the examples are in their attribute space. An example is called incoherent within a dataset if it does not overlap with any other example in a predefined number of attributes.

iii) instance_minimality: An example is subsumed by another example if its attributes form a true subset of another example with the same label. It is useful mostly for relational representations. An example that is not subsumed by another example is minimal.

iv) instance_similarity: The overall similarity of examples in a dataset is defined as the normalized weighted sum of four different local similarity measures

v) instance_uniqueness: An example is unique if and only if there does not exist another identical example.

d) **landmarker:** A different way of probing the structure of the concepts hidden in the data is running some algorithms on it with very different biases, and see how well they perform: the better they do, the closer their bias fits the data. This is what is done naturally when we manually seek an appropriate algorithm: we first select a wide range of very different algorithms, do some preliminary evaluations, and then we remove all algorithms that dont seem to perform well, leaving us with a small set of candidates to evaluate in detail.

i) <u>1-nearest_neighbor_landmarker:</u> Define a distance on the instance space, e.g. the euclidean distance, and label new observation with the observation of the closest training example. In classification problems, the goal of this landmark learner is to determine how close instances belonging to the same class are.

ii) <u>decision_stump_landmarker:</u> Using a decision tree learner, C5.0 to be precise, a single decision node is constructed (representing a single split of the data) which is then to be used for classifying test observations. The goal of this landmark learner is to establish closeness to linear separability.

iii) <u>elite_1-nearest_neighbor_landmarker:</u> A 1-Nearest Neighbor is used again, but only on a subset of attributes, i.e. the most informative attributes according to the information gain ratio. It intents to establish whether a task involves parity-like relationships between its attributes, which means that no single attribute is considerably more informative than another.

iv) <u>linear_discriminant_landmarker:</u> A single linear target function is computed splitting the instance space in two. Like decision stumps, it also establish closeness to linear separability, but not axis-parallel as is the case in the former.

v) <u>naive_Bayes_landmarker:</u> A simple learning algorithm using Bayes theorem to calculate the possibility that an observation belongs to a certain class. Since it assumes that the attributes are conditionally independent from each other, this landmarker is used to measure the extent to which the attributes actually are independent given the class.

vi) <u>random_tree_landmarker:</u> Also using decision trees, an attribute is chosen randomly at each node until the entire tree is built. The goal of this landmark learner is to inform about irrelevant attributes.

vii) <u>worst_node_landmarker:</u> By using the decision trees information gain ratio again, the least informative attribute is used to make the single split. Together with the first landmark learner, this landmarker is supposed to inform about linear separability.

e) **model-based_property:** model-based characterization, builds a model that is typically very fast to induce, and characterizes the data based on properties of that model without doing a full-fletched evaluation of a wide range of learning algorithms.
   i) **decision_tree-based_property:**
      1) distribution_of_branch_lengths
      2) distribution_of_feature_occurrence_in_nodes_
      3) distribution_of_number_of_nodes_per_level
      4) number_of_leafs
      5) number_of_nodes
      6) tree_depth
      7) tree_width

f) **simple_dataset_property:**
   i) dimensionality_of_the_data: ratio of number_of_attributes and number_of_instances
   ii) number_of_binary_features
   iii) number_of_features
   iv) number_of_instances
   v) number_of_instances_with_missing_values
   vi) number_of_missing_values
   vii) number_of_nominal_features
   viii) number_of_numerical_features
   ix) number_of_target_classes
   x) percentage_of_features_with_outliers
   xi) percentage_of_missing_values
   xii) percentage_of_nominal_features
   xiii) percentage_of_numerical_features
   xiv) presence_of_outliers_in_target

g) **statistical_dataset_property**
   i) average_correlation_to_target: Measures the correlation between a numerical attribute X and a numerical target Y_{target}.
   ii) average_feature_kurtosis
   iii) average_feature_skewness
   iv) average_p-value_for_target: measures the correlation between a nominal attribute X and a numerical target Y_{target}.

v) <u>Box's_M_statistic:</u> Box's M-statistic measures the equality of the covariance matrices S_i of the different classes. If they are equal, then linear discriminants could be used, otherwise, quadratic discriminant functions should be used instead. As such, M predicts whether a linear discriminant algorithm should be used or not. In the following, $S_i = \frac{S_{c_i}}{n_i - 1}$ is the i class covariance matrix with S_{c_i} the i class scatter matrix and n_i the number of examples pertaining to class i, and $S = \frac{1}{n-cl}\sum_i S_{c_i}$ the pooled covariance matrix. It is zero when all individual covariance matrices are equal to the pooled covariance matrix.

$$M = \gamma \sum_i (n_i - 1) log \frac{|S|}{|S_i|} \tag{4}$$

$$\gamma = 1 - \frac{2num^2 + 3num - 1}{6(num+1)(cl-1)} (\sum_i \frac{1}{n_i - 1} - \frac{1}{n - cl}) \tag{5}$$

vi) <u>first_canonical_correlation:</u> the first canonical correlation coefficient measures the association between all numerical attributes and a nominal (class) attribute. In principal component analysis (PCA), datasets are transformed into a new dataset with fewer dimensions (attributes). The first dimension, called the first principal component is a new axis in the direction of maximum variance. The variance of this principal axis is given by the largest eigenvalue λ_1. It thus measures how well the classes can be separated by the numerical attributes.

$$\rho_{max} = \sqrt{\frac{\lambda_1}{1 + \lambda_1}} \tag{6}$$

vii) <u>frac1:</u> the fraction of the total variance retained in the 1-dimensional space defined by the first principal component can be computed as the ratio between the largest eigenvalue λ_1 of the covariance matrix S and the sum of all its eigenvalues:

$$frac1 = \frac{\lambda_1}{\sum_i \lambda_i} \tag{7}$$

viii) <u>SD-ratio:</u> SD-ratio, the standard deviation ratio, is a reexpression of Box's M-statistic M which is one if M is zero and strictly greater than one if the covariances differ.

$$SD\text{-}ratio = exp(\frac{M}{num \sum_i (n_i - 1)}) \tag{8}$$

- ix) stationarity_of_target: Indicates whether the standard deviation of the target feature is larger that its mean
- x) target_feature_kurtosis
- xi) target_feature_skewness

- h) **sub-sampling_landmarker:** Runs the learning algorithms on a small sample of the data, which for most algorithms will result in much faster training times. Indeed, learning algorithms typically learn the most from the first few examples. More examples will further fine-tune the model, but the performance gains will be small in comparison with the first few examples. When plotting the performance of learning algorithms on increasingly larger samples of a dataset, the resulting curve is called a learning curve. It will typically shoot up after a small percentage of the data and will, depending on the learning algorithm, start to level off shortly after that. The assumption of subsampling landmarking is that the performance of one point in the beginning of the learning curve will help us to predict the performance on the entire dataset.
  - i) partial_learning_curve
  - ii) single_subsample
- i) **task-specific_dataset_property:**
  - i) clustering_dataset_property
  - ii) **time_series_analysis_dataset_property:**
    - 1) coefficient_of_variation
    - 2) mean_of_first_5_autocorrelations
    - 3) statistical_significance_of_autocorrelations

- II) **feature_property:**
  - 1) **qualitative_feature_property:**
    - a) **feature_datatype:**
      - i) nominal_datatype
      - ii) **numerical_datatype:**
        - 1) boolean_datatype
        - 2) integer_datatype
        - 3) **real_datatype:**
          - a) real_from_0_to_1_datatype
  - 2) **quantitative_feature_property:**
    - a) feature_entropy: the entropy of a nominal attribute X is a measure of the uncertainty (or randomness) associated with

it. It measures the average information content one is missing when one does not know the exact value of X. If entropy is zero (if all values are the same), the attribute contains no information. The class entropy H(C) is the amount of information required to specify the class of an instance, a measure for how informative the attributes need to be. A low H(C) means that the distribution of examples among classes is very skewed (containing some very infrequent classes) which some algorithms cannot handle well.

$$H(X)_{norm} = \frac{H(X)}{log_2 n} \tag{9}$$

Although a definition exists for numerical distributions (using an integral instead of a summation), it is of no use for empirical data, and the entropy of numerical attributes (or targets) is calculated by discretizing the values in equal-length intervals.

b) feature_kurtosis: β, the kurtosis or fatness of the distribution's tail, or the 4th standardized moment

$$\beta = \frac{E(X - \mu_X)^4}{\sigma_X^4} \tag{10}$$

c) **feature_redundancy_property:** Determines the degree of redundancy in a dataset: if two or more attributes are dependent, they don't add much information and only increase the dimensionality of the dataset. This is measured by estimating the strength of the relationship between attributes.

   i) feature_concentration_coefficient: τ_{XY}, the *concentration coefficient* measuring the association between two nominal attributes, or the proportional reduction in the probability of an incorrect guess predicting Y, with J distinct values, using X, with I distinct values

$$\tau_{XY} = \frac{\sum_i \sum_j \frac{\pi_{ij}^2}{\pi_{i+}} - \sum_j \pi_{+j}^2}{1 - \sum_j \pi_{+j}^2} \tag{11}$$

   ii) feature_correlation_coefficient: ρ_{XY}, the correlation coefficient measuring the association between two numerical attributes

$$\rho = \frac{\sigma_{XY}}{\sqrt{\sigma_X^2 \sigma_Y^2}} \tag{12}$$

iii) multiple_correlation_coefficient: R_i, the multiple correlation coefficient measuring the maximal correlation coefficient between a numerical attribute X_i and some linear combination of all other numerical attributes $Z_i\alpha$, with $Z_i = (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_{num})$ and α a non-zero vector.

$$R_i = argmax_{\alpha \neq 0} \frac{\sigma_{X_i Z_i \alpha}}{\sqrt{\sigma^2_{X_i} \sigma^2_{Z_i \alpha}}} \tag{13}$$

iv) mutual_information: $MI(Y, X)$, the *mutual information* between nominal attributes X and Y describes the reduction in uncertainty of Y due to the knowledge of X, and leans on the conditional entropy $H(Y|X)$. It is also the underlying measure of the information gain metric used in decision tree learners.

$$\begin{aligned} MI(Y, X) &= H(Y) - H(Y|X) && (14) \\ H(Y|X) &= \sum_i p(X = x_i) H(Y|X = x_i) && (15) \\ &= -\sum_i \pi_{i+} \sum_j \pi_{j|i} log_2(\pi_{j|i}) && (16) \end{aligned}$$

v) p-value_of_F_distribution: $p_{val_{XY}}$, the p-value of the F-distribution for a nominal attribute X with I values and a numeric attribute Y. The Analysis of Variance (ANOVA) examines how a numerical variable affects a nominal one by examining whether the means of the I groups defined on Y by X are different. The ratio of the *between group variance* and the *within group variance* $MS(B)/MS(W)$ follows the F-distribution and the p-value of that distribution gives the probability of observing that ratio under the assumption that the group means are equal. A p-value close to one means we can accept that assumption, an indication that X heavily affects Y

vi) uncertainty_coefficient: $UC(X, Y)$, the uncertainty coefficient is the mutual information between an attribute X and target attribute Y divided by the entropy of Y. It measures the proportional reduction in the *variance* of Y when X is known. It is strongly related to the *information gain ratio* used in decision trees, which is defined as $UC(Y, X)$, or the proportional reduction in in the variance of X when target

Y is known.

$$UC(X,Y) = \frac{MI(Y,X)}{H(X)} \tag{17}$$

d) feature_skewness: γ, the skewness or lack of symmetry in the distribution, or the 3rd standardized moment

$$\gamma = \frac{E(X-\mu_X)^3}{\sigma_X^3} \tag{18}$$

e) **nominal_feature_properties:**

   i) number_of_feature_values

f) normalized_feature_entropy: $H(X)_{norm}$, the normalized entropy of a nominal attribute X rescales entropy to the [0..1] interval

$$H(X)_{norm} = \frac{H(X)}{log_2 n} \tag{19}$$

g) **numerical_feature_properties:**

   i) maximum_feature_value

   ii) minimum_feature_value

III) **instance_property:**

1) **labeling:**

   a) labeled

   b) unlabeled

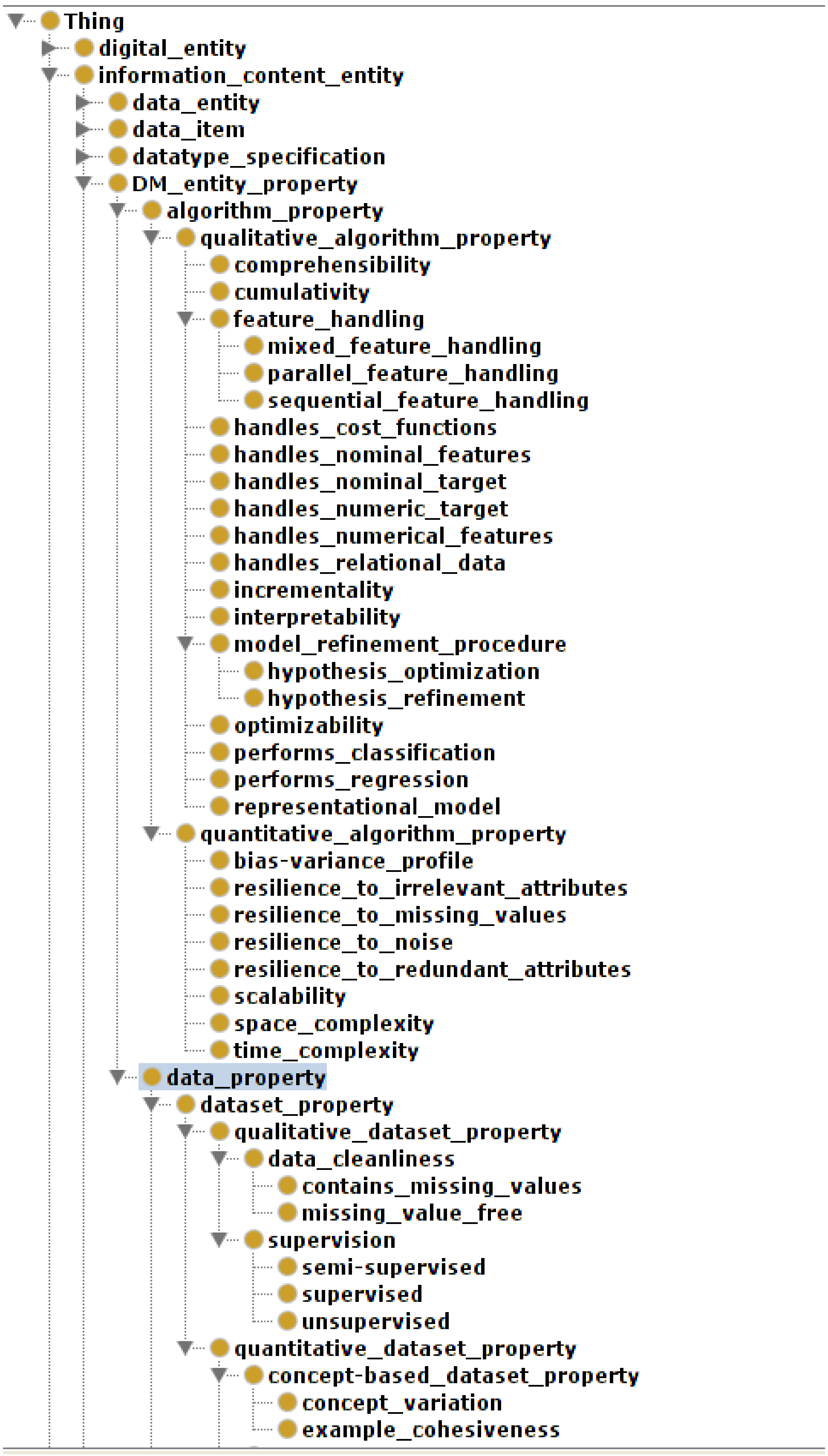
2) **machine_property:**

A) amount_of_memory

B) benchmark_speed

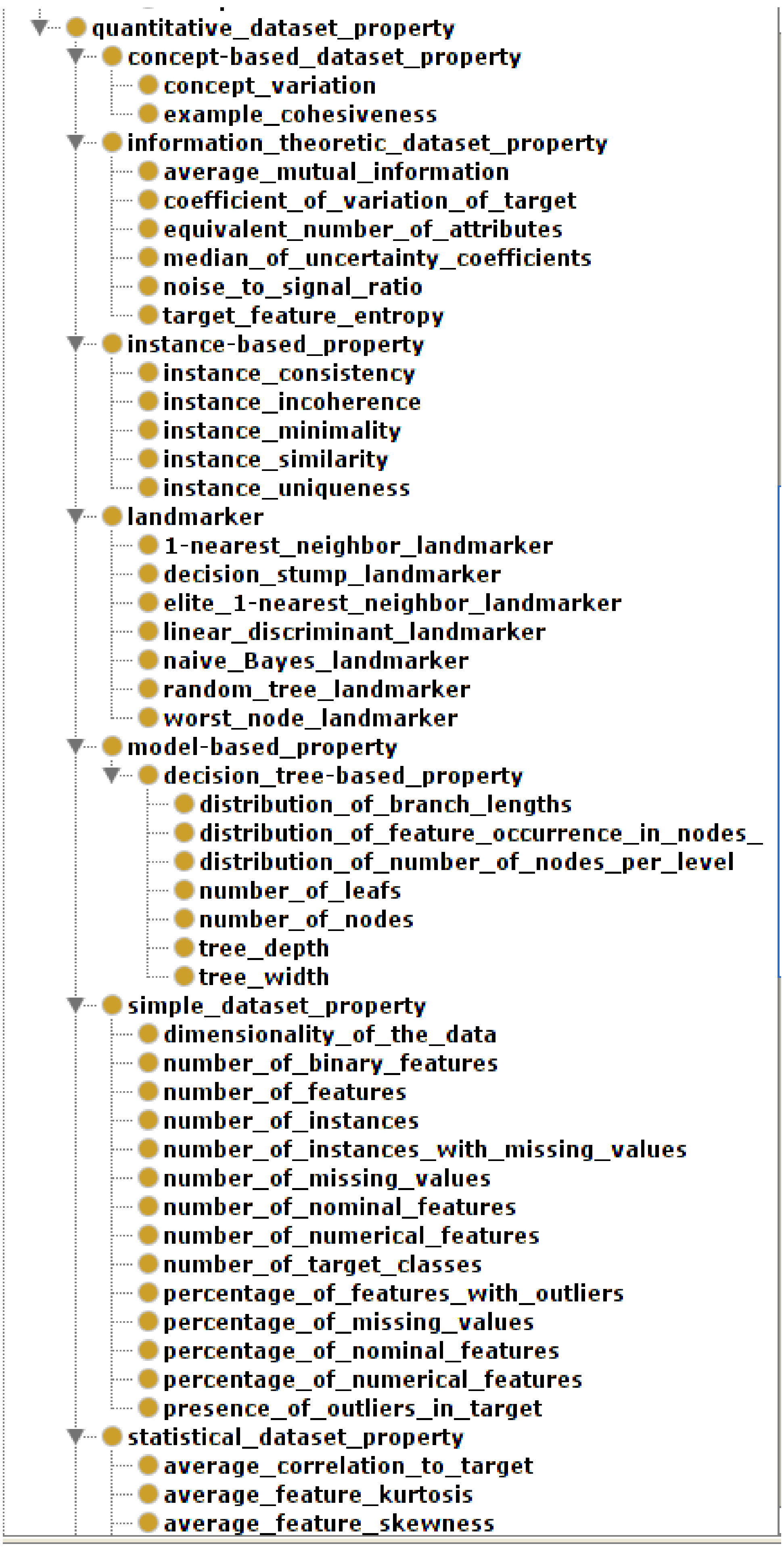
C) cpu_speed

D) number_cpus



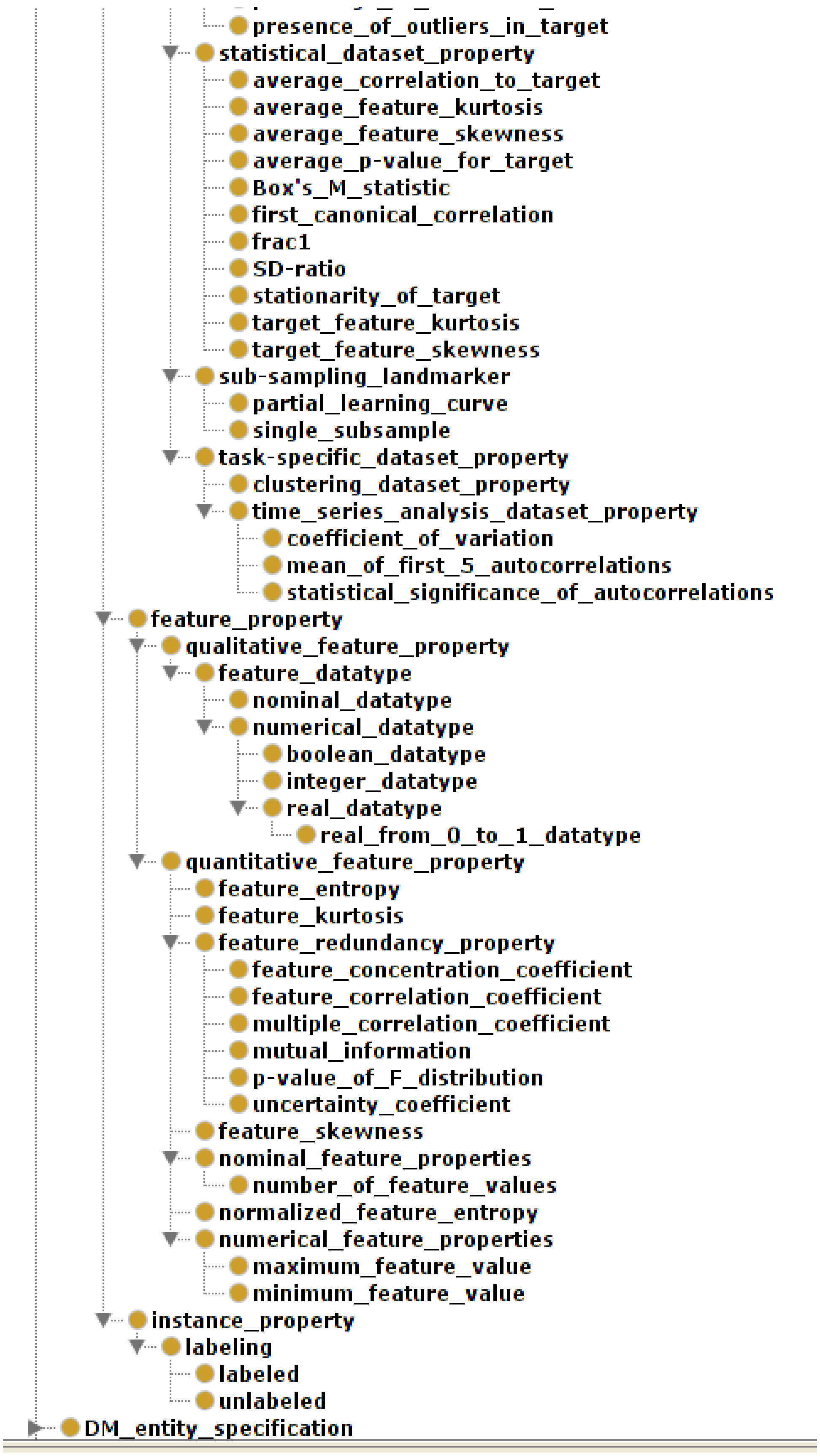
```
Thing
  digital_entity
  information_content_entity
    data_entity
    data_item
    datatype_specification
    DM_entity_property
      algorithm_property
        qualitative_algorithm_property
          comprehensibility
          cumulativity
          feature_handling
            mixed_feature_handling
            parallel_feature_handling
            sequential_feature_handling
          handles_cost_functions
          handles_nominal_features
          handles_nominal_target
          handles_numeric_target
          handles_numerical_features
          handles_relational_data
          incrementality
          interpretability
          model_refinement_procedure
            hypothesis_optimization
            hypothesis_refinement
          optimizability
          performs_classification
          performs_regression
          representational_model
        quantitative_algorithm_property
          bias-variance_profile
          resilience_to_irrelevant_attributes
          resilience_to_missing_values
          resilience_to_noise
          resilience_to_redundant_attributes
          scalability
          space_complexity
          time_complexity
      data_property
        dataset_property
          qualitative_dataset_property
            data_cleanliness
              contains_missing_values
              missing_value_free
            supervision
              semi-supervised
              supervised
              unsupervised
          quantitative_dataset_property
            concept-based_dataset_property
              concept_variation
              example_cohesiveness
```

Figure 1



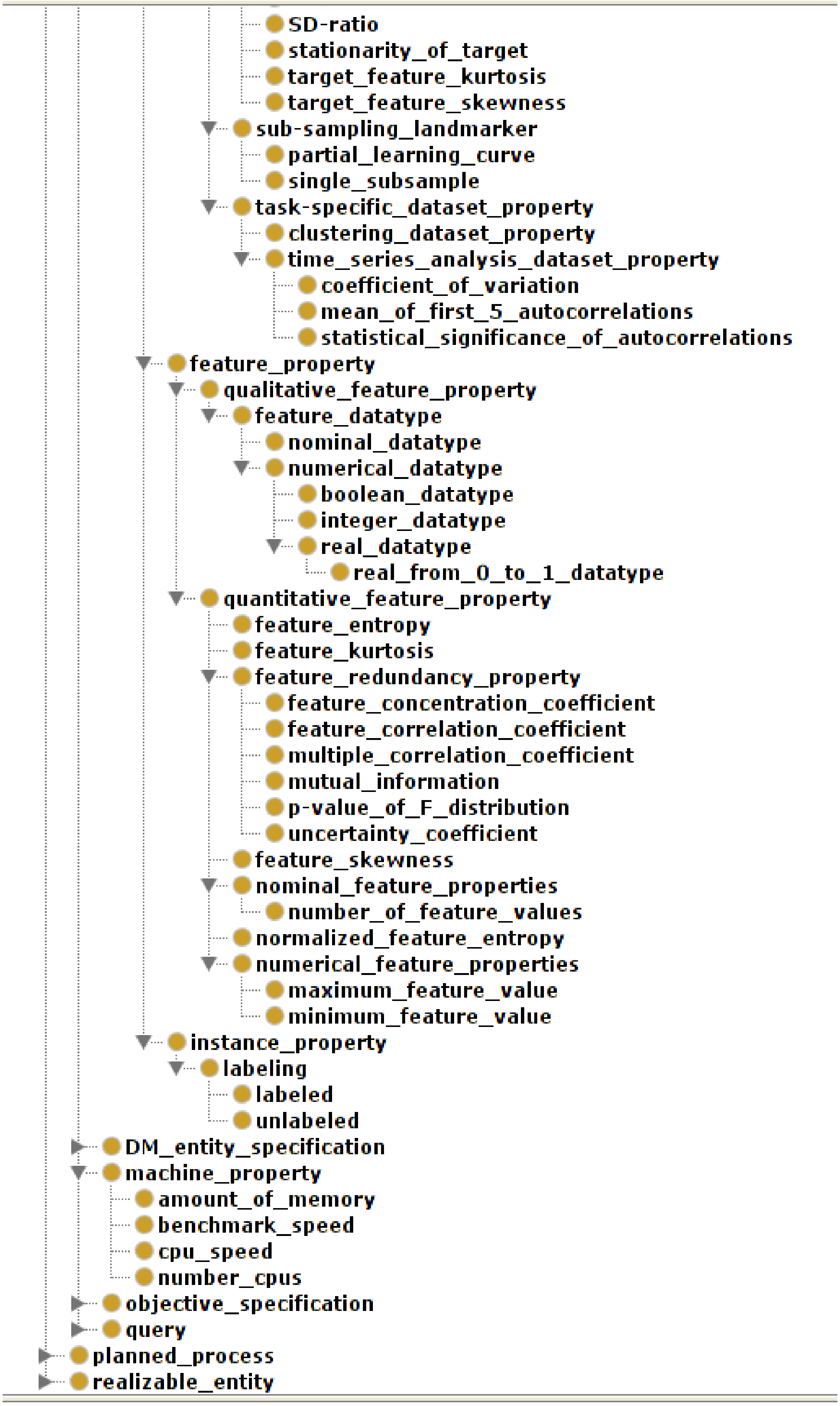
- quantitative_dataset_property
  - concept-based_dataset_property
    - concept_variation
    - example_cohesiveness
  - information_theoretic_dataset_property
    - average_mutual_information
    - coefficient_of_variation_of_target
    - equivalent_number_of_attributes
    - median_of_uncertainty_coefficients
    - noise_to_signal_ratio
    - target_feature_entropy
  - instance-based_property
    - instance_consistency
    - instance_incoherence
    - instance_minimality
    - instance_similarity
    - instance_uniqueness
  - landmarker
    - 1-nearest_neighbor_landmarker
    - decision_stump_landmarker
    - elite_1-nearest_neighbor_landmarker
    - linear_discriminant_landmarker
    - naive_Bayes_landmarker
    - random_tree_landmarker
    - worst_node_landmarker
  - model-based_property
    - decision_tree-based_property
      - distribution_of_branch_lengths
      - distribution_of_feature_occurrence_in_nodes_
      - distribution_of_number_of_nodes_per_level
      - number_of_leafs
      - number_of_nodes
      - tree_depth
      - tree_width
  - simple_dataset_property
    - dimensionality_of_the_data
    - number_of_binary_features
    - number_of_features
    - number_of_instances
    - number_of_instances_with_missing_values
    - number_of_missing_values
    - number_of_nominal_features
    - number_of_numerical_features
    - number_of_target_classes
    - percentage_of_features_with_outliers
    - percentage_of_missing_values
    - percentage_of_nominal_features
    - percentage_of_numerical_features
    - presence_of_outliers_in_target
  - statistical_dataset_property
    - average_correlation_to_target
    - average_feature_kurtosis
    - average_feature_skewness

Figure 2



```
                presence_of_outliers_in_target
            statistical_dataset_property
                average_correlation_to_target
                average_feature_kurtosis
                average_feature_skewness
                average_p-value_for_target
                Box's_M_statistic
                first_canonical_correlation
                frac1
                SD-ratio
                stationarity_of_target
                target_feature_kurtosis
                target_feature_skewness
            sub-sampling_landmarker
                partial_learning_curve
                single_subsample
            task-specific_dataset_property
                clustering_dataset_property
                time_series_analysis_dataset_property
                    coefficient_of_variation
                    mean_of_first_5_autocorrelations
                    statistical_significance_of_autocorrelations
    feature_property
        qualitative_feature_property
            feature_datatype
                nominal_datatype
                numerical_datatype
                    boolean_datatype
                    integer_datatype
                    real_datatype
                        real_from_0_to_1_datatype
        quantitative_feature_property
            feature_entropy
            feature_kurtosis
            feature_redundancy_property
                feature_concentration_coefficient
                feature_correlation_coefficient
                multiple_correlation_coefficient
                mutual_information
                p-value_of_F_distribution
                uncertainty_coefficient
            feature_skewness
            nominal_feature_properties
                number_of_feature_values
            normalized_feature_entropy
            numerical_feature_properties
                maximum_feature_value
                minimum_feature_value
    instance_property
        labeling
            labeled
            unlabeled
DM_entity_specification
```

Figure 3



```
            SD-ratio
            stationarity_of_target
            target_feature_kurtosis
            target_feature_skewness
          sub-sampling_landmarker
            partial_learning_curve
            single_subsample
          task-specific_dataset_property
            clustering_dataset_property
            time_series_analysis_dataset_property
              coefficient_of_variation
              mean_of_first_5_autocorrelations
              statistical_significance_of_autocorrelations
      feature_property
        qualitative_feature_property
          feature_datatype
            nominal_datatype
            numerical_datatype
              boolean_datatype
              integer_datatype
              real_datatype
                real_from_0_to_1_datatype
        quantitative_feature_property
          feature_entropy
          feature_kurtosis
          feature_redundancy_property
            feature_concentration_coefficient
            feature_correlation_coefficient
            multiple_correlation_coefficient
            mutual_information
            p-value_of_F_distribution
            uncertainty_coefficient
          feature_skewness
          nominal_feature_properties
            number_of_feature_values
          normalized_feature_entropy
          numerical_feature_properties
            maximum_feature_value
            minimum_feature_value
      instance_property
        labeling
          labeled
          unlabeled
  DM_entity_specification
  machine_property
    amount_of_memory
    benchmark_speed
    cpu_speed
    number_cpus
  objective_specification
  query
planned_process
realizable_entity
```

Figure 4