

Interpretable Taxonomy Extraction from Digital Assets Metadata Using Automated Unsupervised Decision Tree Learning

Ildar Baimuratov¹[0000–0002–6573–131X]

ITMO University, Saint-Petersburg, Russia baimuratov.i@gmail.com

Abstract. This research is aimed at interpretable taxonomy extraction from digital assets metadata using automated unsupervised decision tree learning. The method proposed is based on automated unsupervised decision tree learning. It has the following advantages: i) it allows extracting taxonomies from various data types including numerical, categorical, and textual data, ii) as it uses a decision tree, the method allows classifying digital assets into extracted taxons, iii) the extracted taxonomy is interpretable in terms of the presence or the absence of tokens in assets metadata, iv) the method does not require any extra data, any labeled assets, or v) any manual hyperparameter optimization. Moreover, the extracted taxonomy and classified assets are converted into a knowledge graph for further representation and usage in digital asset management. The method is prototyped, tested on several collections of digital assets, and its usage is demonstrated on one of them.

Keywords: Taxonomy extraction · Digital Asset Management · Unsupervised Learning · Interpretable clustering · Decision Tree · Hyperparameter Optimization · Knowledge graph.

1 Introduction

Taxonomies provide search, navigation, and management of digital assets. On the other hand, manually creating and maintaining taxonomies is extremely labor intensive, and without proper support, digital asset taxonomies quickly become irrelevant and useless. Moreover, manually constructed taxonomies, as opposed to data-driven ones, can be subjective and biased. Thus, data-driven taxonomy extraction is tends to be required.

Most of existing approaches to taxonomy extraction do it from text corpora. However, not all types of digital assets are text-rich, for example, images are not. In general, digital assets have only metadata available. Extracting taxonomies from metadata is challenging. Unlike texts, metadata commonly do not contain explicitly enough words from which it would be possible to extract relationships between concepts. There are approaches for taxonomy extraction that use metadata, however, they appear to require some extra data like general purpose texts [5], or a user-provided base taxonomy [7]. Augmenting data demands

some preliminary investment, this fact narrows the applicability of the suggested approaches.

Therefore, the present research is aimed at taxonomy extraction from digital assets metadata. The taxonomy extraction process is also required to be automated and unsupervised, and the extracted taxonomies have to be interpretable. To achieve the above goal, the following tasks have been set: i) to review the existing methods of taxonomy extraction; ii) to develop a method for taxonomy extraction from digital assets metadata; iii) to implement a prototype based on the developed method; iv) to test the developed prototype on a collection of digital assets.

The proposed solution is based on automated unsupervised decision tree learning. This method has several advantages: i) it allows extracting taxonomies from data of mixed types, including numerical, categorical, and textual ones, ii) as it uses a decision tree, the method allows classifying digital assets into extracted taxons, iii) the extracted taxonomy is interpretable in terms of the presence or absence of tokens in assets metadata for categorical or textual features, or in terms of comparing to a threshold for numerical ones, iv) being unsupervised, the method does not require any additional data or labeled assets, v) being automated, the method does not require manual hyperparameter optimization. Moreover, the extracted taxonomy and classified assets can be converted into a knowledge graph for further representation and processing.

The contribution of this research is as follows: i) a method for taxonomy extraction from metadata solely, ii) a method for taxonomy construction from a decision tree, iii) a method for automated unsupervised decision tree learning, iv) application of the taxonomy extraction method to digital asset management.

The outline of the paper is as follows: the second section discusses research related to taxonomy extraction from digital assets metadata based on automated unsupervised decision tree learning, the third section describes the proposed method of taxonomy extraction, including preprocessing, automated unsupervised tree learning, digital assets classification, and conversion of the decision tree and the classified assets into a knowledge graph, the fourth section refers to prototyping the proposed method, and applying it to a collection of digital assets, including a description of data, prototype, and results of testing.

2 Related work

The present section provides a review of research related to interpretable taxonomy extraction from digital assets metadata using automated unsupervised decision tree learning. First, consider the existing solutions suggested for taxonomy extraction. For each solution, it has to be checked whether metadata solely is sufficient for taxonomy extraction, whether the taxonomy extraction is interpretable, unsupervised, and automated.

Article [8] presents a method for taxonomy construction from keywords and from a text corpus. It is based on heuristic rules and syntactic parsing. It has been applied to the environmental protection. The authors of [3] form a taxon-

omy from a graph whose nodes correspond to terms, and whose edges correspond to the degree of their semantic similarity. The input data is a corpus and a set of terms. Hierarchical relationships are generated using the Monte Carlo sampling and selected by the maximum likelihood. The solution presented in [5] aims at generating taxonomies based on a set of key phrases. Hierarchical clustering is used, however, to extract the relationships between the key phrases, the data is enriched with the context from a corpus of general purpose texts. In [7], taxonomies are built from a user-provided base taxonomy and a corpus. The proposed algorithm iteratively expands the base taxonomy wide and deep using the similarity between terms, until the set of entities in the corpus is exhausted. The algorithm applies syntactic parsing and semi-supervised relation extraction. The authors of [10] also pursue the goal of automated taxonomy construction, however, concepts in the taxonomies are required to be represented by a cluster of semantically related terms rather than by one term. The proposed method applies vector representations of the original corpus and iterative centroid clustering. The analysis of the above solutions is presented in Table 1. There does not seem to be a method for interpretable, automated, and unsupervised taxonomy extraction from metadata solely.

Table 1: Taxonomy extraction methods

Solution	Metadata only	Interpretable	Automated	Unsupervised
Tsui et al. [8]	-	-	-	-
Fountain [3] & Lapata	-	-	-	-
Liu et al. [5]	-	-	+	+
Shen et al. [7]	-	-	+	-
Zhang et al. [10]	-	+	+	+

There has also been some previous research on interpretable taxonomy construction from unsupervised trees. In [1], the authors proposed a method for unsupervised decision tree learning. In this method, at any given node an attribute is chosen in such a way that if clustering is solely performed based on that attribute, the resulting inhomogeneity is minimized. Two algorithms for splitting data are presented: 1) valley detection in a histogram gradient, and 2) binary splitting with maximization of inhomogeneity gain. In the proposed method, the minimum number of data points at a node was used as the stopping criterion. No optimization of this hyperparameter is suggested. Another method for unsupervised decision tree learning is proposed in [4]. The method includes three steps: maximal tree construction, its pruning using a dissimilarity criterion, and joining similar, but not adjacent clusters. The algorithm depends on the hyperparameters of minimum deviance reduction for splitting, and the minimum cluster size. No optimization of these hyperparameters is proposed either. Thus, there appears to be no method for automated unsupervised tree learning.

3 Method

The proposed approach to interpretable taxonomy extraction for digital assets consists of binary vectorization of assets metadata, automated unsupervised decision tree learning, and converting the tree into a taxonomy. Besides, the learned decision tree is applicable for forming recommendations on digital assets classification, and the extracted taxonomy with the classified assets can be converted into a knowledge graph. To specify, taxonomy extraction pipeline includes the following steps:

1. preprocessing:
 - (a) feature selection from digital assets metadata,
 - (b) binary vectorization of non-numerical selected features,
2. automated unsupervised decision tree learning:
 - (a) generating unique labels for digital assets,
 - (b) decision tree learning using the vectorized features and generated labels,
 - (c) optimizing the depth of the learned decision tree through a clustering quality gain,
3. forming recommendations on classifying digital assets using the optimized decision tree,
4. converting the decision tree and the classified assets into a knowledge graph.

3.1 Preprocessing

For performance improvement, a selection of informative features is executed. For each feature, the information entropy is calculated, and features with an entropy lower than a specified threshold are removed. The threshold is determined empirically.

For the CART [2] decision tree learning algorithm considered in this research all features must be numerical. Thus, non-numerical features have to be vectorized. The method for vectorization depends on the feature type. For categorical features, one-hot encoding is used. Text features are vectorized using the binary bag of words method. Sequences of 2 or more alphanumeric characters are considered as tokens, punctuation is ignored and treated as a token separator. Stop words are dropped out. After vectorization, original non-numerical features are removed, and the obtained vectors are appended to the data set.

For vectorization of text features, TF/IDF could be considered as an option, however, with the TF/IDF vectorization, the rule corresponding to the vectorized feature will have the structure 'Token > x ', where the numeric value x does not have any meaning to a user. On the other hand, with binary vectorization, the rules have the form 'token' or 'no token', which means that metadata of the corresponding digital assets contain or do not contain the specified token. Thus, binary vectorization was chosen, since it allows forming more interpretable taxon descriptions.

The proposed method also includes applying clustering quality metrics. For correct clustering quality evaluation, numerical features have to be consolidated

with others. For this purpose, as other features are binary, normalization is applied to numerical features.

However, using numerical features has a drawback. A taxon formed from rules for binary features is described by a single token, while a taxon formed from rules for numeric features is described by an expression of the form 'feature $> x$ ', which may be less interpretable for taxonomy users. Besides, the decision tree learning algorithm tends to use numerical features more often than binary, because they faster decrease an impurity. To avoid the prevalence of poorly interpretable taxons, numerical features can be discretized and one-hot encoded, similar to categorical ones. For example, the 'PhotoDate' feature from the considered data set can be discretized by years, then taxons of the form 'PhotoDate $> yyyy-mm-dd\ hh:mm:ss$ ' would be replaced with taxons of the form 'yyyy'.

3.2 Automated unsupervised decision tree learning

A decision tree is used as a base for taxonomy construction. Similar to hierarchical clustering algorithms, the decision tree forms a tree-like model for objects distribution. However, the decision tree has the advantage that it explicitly describes a taxon using the condition, by which an asset is assigned to the taxon. Together with binary vectorized features, such conditions take the form of the presence or absence of the corresponding token in the asset's metadata.

However, the decision tree is a supervised learning algorithm, hence it requires labeled data for training. To overcome this limitation, it is proposed to train the algorithm in a self-supervised manner, generating a unique label for each asset. With this approach, optimization of the decision tree depth becomes crucial, because otherwise, each asset will have its own taxon.

In this research, the decision tree depth is optimized through the minimum impurity decrease threshold. As the unsupervised approach is applied, the impurity decrease threshold can not be optimized using accuracy as an objective function. On the other hand, after each split a decision tree forms a partition of objects, which can be considered as a clustering. Thus, instead of accuracy, it is proposed to optimize the decision tree depth using a clustering quality index, like Silhouette score [6], to give an example. Equation (1) defines sil_i as the silhouette score sil for the dataset D and predictions of a decision tree DT trained on D and unique labels l using minimum impurity decrease threshold t_i :

$$sil_i = sil(D, DT(D, l, t_i)). \quad (1)$$

Given the objective function, the minimum impurity decrease threshold is optimized with the elbow rule [9]. The elbow rule is used as the optimization strategy instead of global clustering quality maximization, because increasing the decision tree depth reduces the uncertainty in the partition of objects, therefore, with the depth of the tree, the clustering quality increases nearly monotonously. Thus, global clustering quality maximization would lead to a trivial optimum equal to the minimum value for the impurity decrease threshold from the search space. On the contrary, the elbow corresponds to the impurity decrease threshold, after which the clustering quality gain starts decreasing.

To find the elbow, at each iteration, a decision tree is built using a threshold t_i , and the silhouette gain $sil_i - sil_{i-1}$ compared to the previous iteration is calculated. For the first iteration, this gain is set to zero. Iteration stops if the number of labels predicted with the tree makes one, in other words, if the tree produces only one cluster. As a result, the dependency of the silhouette score gain on the minimum impurity decrease threshold is formed. The optimal impurity decrease threshold t^* is defined as the threshold that gives the maximum silhouette score gain:

$$t^* = \arg \max_i (sil_i - sil_{i-1}).$$

3.3 Assets classification

The learned decision tree is naturally used for classifying digital assets. In this research, it is proposed to enhance the classification with a confidence coefficient, forming a recommendation system for digital asset management. The confidence coefficient provides better interpretability of the assets classification and mitigates possible classification errors. The assets classified with a low confidence coefficient are supposed to be revised by a user.

As the tree is learned from synthetic unique labels, confidence coefficients based on a probability of classifying an object into a specific class, common for decision trees, become meaningless. Thus, for an unsupervised decision tree, the classification confidence coefficient is formed similar to clustering. Namely, the confidence $conf(x, T)$ of a recommendation to assign an asset x to a taxon T is calculated as the distance $d(x, c)$ from the point x to the centroid c of the cluster corresponding to the taxon T . For better interpretability, the distance is normalized and inverted. The distance is normalized regarding the maximum distance for a given cluster, that is, the distance between the centroid and its farthest point:

$$conf(x, T) = 1 - \frac{d(x, c)}{1 + \max_i(d(x_i, c))},$$

where $c = mean(T)$. Thus, the confidence of a recommendation to assign an asset to a taxon is 1, when the asset is the centroid of the corresponding cluster, and the confidence is ≈ 0 , when the asset is the farthest from the centroid.

3.4 Knowledge graph construction

For further representation and processing, the learned decision tree and the classified assets are converted into a knowledge graph. Namely, the decision tree is converted into a hierarchy of OWL-classes, and the classified assets are converted into individuals.

The algorithm for converting nodes of the decision tree into OWL-classes is as follows:

- for each node, the OWL-class is created for each of its children,
- the existing class corresponding to the parent node is set to be the child's superclass,

- if the parent node is the root of the tree, the child is set to be a subclass of the 'Thing' class,
- if a feature considered at the parent-node has the float type, the entire rule is used as a label for the created OWL-class, thus, the label has the form 'feature $> x$ ' or 'feature $\leq x$ ',
- if the parent's feature is binary and the child node is accessed by a rule of the form 'feature > 0.5 ', the child node is labeled with the feature name itself,
- otherwise, if the child node is accessed by a rule of the form 'feature ≤ 0.5 ', it is labeled with the feature name with the prefix 'no '.

After constructing the hierarchy of the OWL-classes, for each classified asset, an individual is created. It is assigned to a class that corresponds to the leave of the decision tree, to which the asset has been classified. The classification confidence coefficient is specified as a DataProperty.

The CART algorithm considered in this research produces only binary trees. Being binary seems to be an unnatural and strong limitation on the extracted taxonomies. This limitation can be avoided in one of the two ways. First, it is possible to consider another decision tree learning algorithm that can produce non-binary trees. Second, it is possible to apply postprocessing to the extracted taxonomies. For example, postprocessing may include removing taxons of the form 'not token', assigning children and instances of the removed taxons to their parents.

4 Experiments

4.1 Data

For an experimental study, four collections of digital assets metadata were used. They have the same structure, but different topics: i) sport events, ii) sport club, iii) business news, and iv) landscapes. Each data set contains about 4000 records and 9 features presented in Table 2.

Table 2: Assets metadata

N	Feature	Type
1	Keywords	Text
2	Headline	Text
3	Description	Text
4	Country	Categorical
5	State	Categorical
6	City	Categorical
7	PhotoDate	Integer
8	CreatorName	Categorical
9	Source	Categorical

4.2 Prototype

The prototype is implemented in Python language. NumPy, SciPy, pandas, scikit-learn, NLTK, and owlready2 libraries are used. The prototype takes a data set in .csv format as input, the preprocessing module performs data normalization, feature selection, and vectorization. The vectors formed are passed into the decision tree module, it learns a decision tree from vectors and uses the tree to classify data. The learned decision tree and the classified assets are passed into the knowledge graph module, which transforms the obtained results into a knowledge graph. The knowledge graph is saved as .rdf file. Fig. 1 is the prototype architecture.

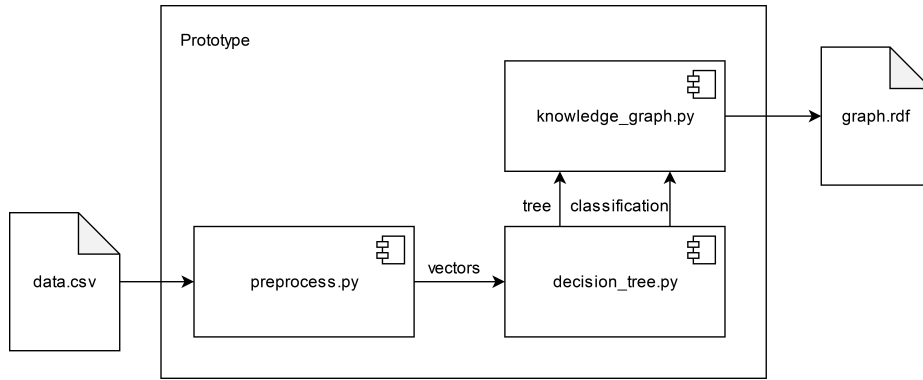


Fig. 1: Prototype architecture

4.3 Preprocessing

An application of the prototype is demonstrated for the collection of digital assets with landscapes. For this set, the text normalization involves removing the escape characters and filling in the missing values.

For feature selection, the information entropy was calculated. Entropy for each feature is presented in Table 3. Features with entropy lower than 25% of the maximum were dropped. This value was chosen based on the empirical studies. For the considered data set, the threshold of 25% dropped features with too many missing values. Normalized entropy of the selected features is presented in Table 4. The features 'Country', 'CreatorName', and 'Source' were dropped.

The data set discussed contains three types of features: numeric, categorical, and textual. For each feature, the corresponding transformation method was applied. Thus, the numeric feature 'PhotoDate' was normalized. The categorical features 'State', and 'City' were one-hot encoded. The text features 'Keywords', 'Headline', and 'Description' were binary vectorized. Due to computational limitations, only unigrams were used for token generation. The vectorized data set contained 4029 features.

Table 3: Metadata entropy

N	Feature	Entropy
1	Keywords	4.46
2	Headline	3.3
3	Description	3.27
4	Country	0.56
5	State	1.38
6	City	2.41
7	PhotoDate	4.13
8	CreatorName	0.72
9	Source	0

Table 4: Metadata after selection

N	Feature	Normalized entropy
1	Keywords	1
2	Headline	0.74
3	Description	0.73
4	State	0.31
5	City	0.54
6	PhotoDate	0.93

4.4 Decision tree learning

The obtained vector set was used to learn a decision tree. First, for each as-set, a unique label was generated. Then, the tree depth was optimized through the minimum impurity decrease. The optimum was searched over the interval $[0.01, 1]$ with the step 0.01. The resulting dependencies of the silhouette score and of its gain on the hyperparameter are shown in Fig. 2. For the data set considered, the silhouette score gain reached the maximum at the impurity reduction threshold equal to 0.24. The decision tree learned with this hyperparameter is presented in Fig. 3. It has 3 layers and 6 leaves.

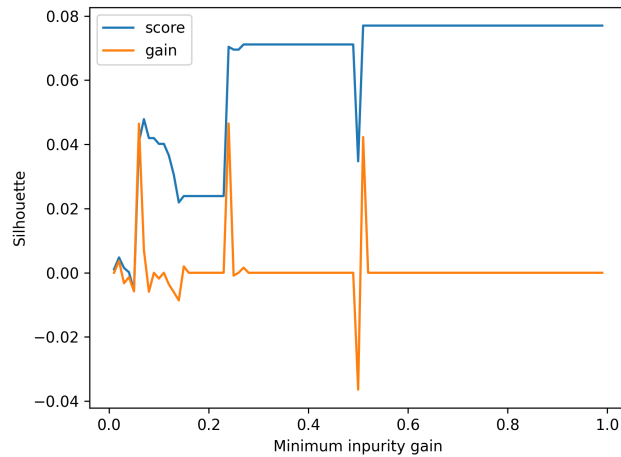


Fig. 2: Hyperparameter optimization

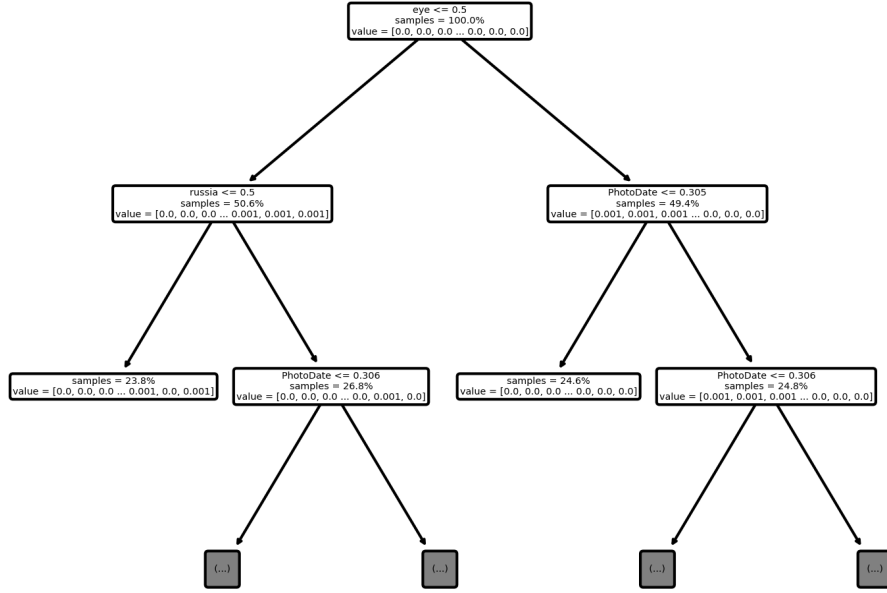


Fig. 3: Decision tree

4.5 Assets classification

The learned decision tree was used to form recommendations on classifying digital assets. The recommendations include confidence coefficient calculated using the silhouette score. Examples of recommendations are presented in Table 5. The distribution of the assets by the taxons is provided in Fig. 4.

Table 5: Examples of recommendations

Asset ID	Taxon	Coef
4804	10	0.19
3209	10	0.2
3400	10	0.17
1516	9	0.54
1566	9	0.52
3452	9	0.47
4751	2	0.56
4403	7	0.46

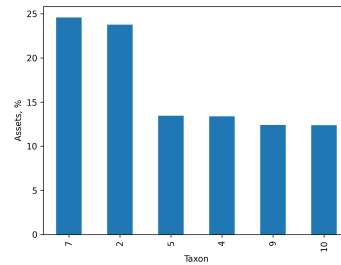


Fig. 4: Distribution of assets by taxons

4.6 Knowledge graph construction

The obtained results are converted into a knowledge graph. The decision tree is transformed into an OWL-classes hierarchy, and the classified assets are transformed into individuals. Fig. 5 represents the resulting OWL-class hierarchy in the Protege interface. In this hierarchy, there are 4 classes extracted from categorical or text features, and 6 classes extracted from the numerical feature 'PhotoDate'. For example, the class 'russia' contains instances of digital assets that have a token 'russia' in their metadata, and the class 'PhotoDate <= 2017-10-03 14:52:28' contains assets that are dated not later than 2017-10-03 14:52:28. Listing 1.1 provides an example of a class in the turtle syntax, and listing 1.2 provides an example of an individual.

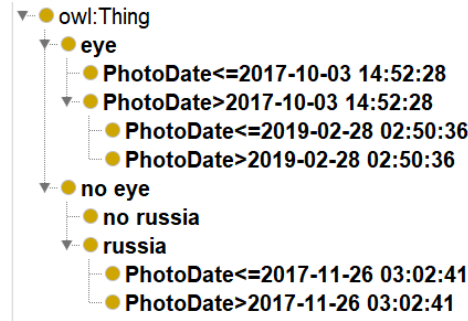


Fig. 5: Taxonomy

Listing 1.1: Class

```
### test#class10
:class10 rdf:type owl:Class ;
         rdfs:subClassOf :class8 ;
         rdfs:label "PhotoDate>2019-02-28 02:50:36"^^xsd:string .
```

Listing 1.2: Individual

```
### test#asset1000
:asset1000 rdf:type owl:NamedIndividual ,
               :class7 ;
           :Coef 0.69 .
```

Finally, the Table 6 provides the run time and the memory usage of this experiment conducted on the Intel Core i5-1035G1 CPU with 8 GB RAM.

5 Conclusion

The aim of this research was to develop a method for taxonomy extraction from digital assets metadata. This method was also required to be automated, un-

Table 6: Run time and memory usage

Step	Time (s)	Memory (MB)
Preprocessing	42	362
Decision tree learning	70	504
Assets classification	6	336
Knowledge graph construction	1	154
Overall	119 (sum)	504 (max)

supervised, and interpretable. A review of existing research showed that there are methods for supervised, semi-supervised, self-supervised, and unsupervised methods for taxonomy extraction from metadata augmented with text corpora or user behavior, but there does not seem to be a method for taxonomy extraction from metadata solely. Thus, a new method was proposed based on a decision tree learned in an automated and unsupervised way. The unsupervised decision tree provides descriptions of generated taxons, interpretable in terms of the presence or the absence of a specified token in the assets metadata. The algorithm for taxonomy extraction includes three main steps. First, the metadata is preprocessed and vectorized as a binary bag of words. Next, the decision tree is learned using unique generated labels and the depth regularization by the silhouette score gain. The learned decision tree is used to form recommendations on classifying digital assets. Finally, the learned decision tree and the classified assets are converted into a knowledge graph for further representation and usage in digital asset management. The method was prototyped and tested on several collections of digital assets. The prototype usage was demonstrated on one of the collections, that of landscapes. Further research will be aimed at extending the method to horizontal relations, resulting in extraction of comprehensive knowledge graphs of digital assets.

Acknowledgements. This research was supported by Picvario LLC¹.

References

1. Basak, J., Krishnapuram, R.: Interpretable hierarchical clustering by constructing an unsupervised decision tree. *IEEE Transactions on Knowledge and Data Engineering* **17**(1), 121–132 (2005)
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA (1984)
3. Fountain, T., Lapata, M.: Taxonomy induction using hierarchical random graphs. pp. 466–476 (06 2012)
4. Fraiman, R., Ghattas, B., Svarc, M.: Interpretable clustering using unsupervised binary trees. *Advances in Data Analysis and Classification* **7** (03 2011). <https://doi.org/10.1007/s11634-013-0129-3>

¹ <https://picvario.com/>

5. Liu, X., Song, Y., Liu, S., Wang, H.: Automatic taxonomy construction from keywords. p. 1433–1441. KDD '12 (2012)
6. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* **20**, 53–65 (1987)
7. Shen, J., Wu, Z., Lei, D., Zhang, C., Ren, X., Vanni, M.T., Sadler, B.M., Han, J.: Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. p. 2180–2189. KDD '18 (2018)
8. Tsui, E., Wang, W., Cheung, C., Lau, A.S.: A concept–relationship acquisition and inference approach for hierarchical taxonomy construction from tags. *Information Processing & Management* **46**(1), 44–57 (2010)
9. Yuan, C., Yang, H.: Research on k-value selection method of k-means clustering algorithm. *J* **2**(2), 226–235 (2019)
10. Zhang, C., Tao, F., Chen, X., Shen, J., Jiang, M., Sadler, B., Vanni, M., Han, J.: Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. pp. 2701–2709 (07 2018)