Concept Tree: High-Level Representation of Variables for More Interpretable Surrogate Decision Trees

Xavier Renard [*1] **Nicolas Woloszko** [*2] **Jonathan Aigrain** [*1] **Marcin Detyniecki** [1 3 4]

Abstract

Interpretable surrogates of black-box predictors trained on high-dimensional tabular datasets can struggle to generate comprehensible explanations in the presence of correlated variables. We propose a model-agnostic interpretable surrogate that provides global and local explanations of black-box classifiers to address this issue. We introduce the idea of *concepts* as intuitive groupings of variables that are either defined by a domain expert or automatically discovered using correlation coefficients. Concepts are embedded in a surrogate decision tree to enhance its comprehensibility. First experiments on FRED-MD, a macroeconomic database with 134 variables, show improvement in human-interpretability while accuracy and fidelity of the surrogate model are preserved.

1. Introduction

The field of interpretability aims at providing users and practitioners with techniques meant to explain either globally a trained machine learning model or locally a particular prediction made by a model. This can be achieved either by training directly an interpretable model, or in a post hoc approach, using model-agnostic or model-specific interpretability techniques.

This paper focuses on post hoc surrogate models that globally approximate a machine learning classifier while providing explanations at the local level of each prediction. We are interested in model-agnostic interpretability approaches meant to be applied on standard feature spaces composed of tabular data. Our goal is to explain any type of trained model: the classifier is a black-box left to the discretion of the practitioners. We refer the reader to recent published surveys for a global picture of the interpretability field as for instance (Guidotti et al., 2018).

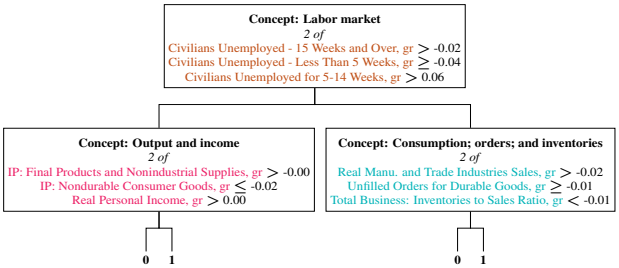


Figure 1. Concept Tree trained on FRED-MD macroeconomic dataset. Variables are grouped by Concepts to constraint the training of an interpretable surrogate decision tree

Surrogate models aiming at providing post hoc interpretability may induce confusion by conveying a false sense of simplicity, especially when subgroups of dependent variables are involved. We refer to dependent variables as variables sharing similar information and possibly generated by a common phenomenon. It may include the various lags of a given time series, various features of a variables, or various measures of a given fact. Surrogate models may arbitrarily select one given variable among a group of dependent variables, thus obscuring the global picture. Subsequently, practitioners may better understand a surrogate model that retains the whole set of dependent variables and depicts a bigger picture than a simpler model.

This paper introduces the idea of *concept*. A *concept* is a representation gathering a group of dependent variables. It can be defined using either domain knowledge or statistical properties of dependent variables (such as the Pearson correlation). The use of *concepts* allows to provide high-level representations that practitioners may find easier to interpret. We contend that *concept*-based methods may be better suited to human understanding and provide more practitioner-friendly representations of a black-box classifier.

We substantiate that claim with an application to decision

[*]Equal contribution [1]AXA, Paris, France [2]OECD, Paris, France [3]Sorbonne Universite, CNRS, LIP6, Paris, France [4]Polish Academy of Science, IBS PAN, Warsaw, Poland. Correspondence to: Xavier Renard <xavier.renard@axa.com>, Nicolas Woloszko <nicolas.woloszko@oecd.org>.

2019 Bank of England and King's College London joint conference on Modelling with Big Data and Machine Learning: Interpretability and Model Uncertainty, London, UK. Copyright by the author(s).

tree surrogates. Decision trees are universally considered interpretable by domain experts (Freitas, 2014). We compare standard surrogate tree models to trees whose training is constrained by the grouping of subgroups of variables in *concepts*. More specifically, we embed the idea of *concept* in the TREPAN algorithm (Craven, 1996), an interpretable decision tree originally instantiating a variant of *id2-of-3* (Murphy & Pazzani, 1991) with a mechanism of oracle querying aiming at populating areas of the training set where the fidelity of the surrogate can be improved. In our approach, the *concepts* are used at each node of the decision tree to constrain the training of the split rule based on *id2-of-3*. We compare the resulting *Concept Trees* to the surrogates provided by the original TREPAN algorithm.

The next section expands on the motivation and formally introduces the idea of *high-level concepts*. Section 3 introduces *Concept Trees*, a version of the TREPAN algorithm that builds on *concepts*, and shows that *Concept Trees* meet the prerequisites of a global-to-local, post-hoc and model-agnostic surrogate. Section 4 assesses both the qualitative and quantitative relevance of our proposition through experiments led on FRED-MD, a monthly macroeconomic database designed for empirical analysis of the US economy (McCracken & Ng, 2016).

2. Concept: Grouping Dependent Variables into High-Level Representation of Variables

It is often the case that groupings of variables in a given dataset may naturally appear. Such grouping can derive from similar meaning or a similar origin (*e.g.* unemployment among men, unemployment among women, unemployment among young people...). A grouping can also be the result of multiple transformations applied to a given source of data (such as multiple lags of a time series, or features engineered from the same variable).

In this work, we consider two types of *concepts*: expert-defined grouping of features and automatically-defined grouping based on a statistical criterion such as feature correlation. Expert-based *concepts* may be used when domain knowledge is available. Automatically-defined concepts do not require prior domain knowledge.

Exploiting the group structure of variables has already been used in the literature to train more accurate sparse models, for instance with *group-lasso* (Yuan & Lin, 2006) or *sparse-group-lasso* (Simon et al., 2013). In the latter, improved accuracy is observed with variable groupings such as gene pathways or factor level indicators in categorical data. Other machine learning fields also cover the idea of grouping dimensions, such as subspace clustering (Vidal, 2011).

In the field of interpretability, the idea of exploiting a meaningful grouping of features to generate better explanations has emerged, for instance with topic-modeling-based feature compression (Kim et al., 2015) or on image classification with deep learning models (Kim et al., 2017; Ghorbani et al., 2019).

Correlated features is a known challenge when building machine learning models and interpreting feature importances (Bühlmann et al., 2013; Gregorutti et al., 2017; Strobl et al., 2008; Tolosi & Lengauer, 2011). For instance, *lasso*-based methods for feature selection tend to select only one representative from a group of correlated features and to discard the others (Bühlmann et al., 2013). It has been pointed out that correlated features severely impact variable importance measures of random forests (Strobl et al., 2008; Gregorutti et al., 2017). Also, many feature selection methods suffer from a *correlation bias*: features belonging to a group of correlated features receive weights inversely proportional to the size of the group (Tolosi & Lengauer, 2011). This issue creates instability in the feature selection process. Small changes in the training data can result in significant changes in the selected set of features. This instability prevents a robust interpretation of variable importance.

We propose to use the idea of *concept* to address both expert-defined grouping of features and automatically (correlated)-defined grouping. *Concepts* are embedded into surrogate models in order to constrain their training, which provides two levels of granularity for the explanations: at high-level (concept) and at finer level (raw variables). The next paragraph offers a formal presentation of the idea of *concepts*.

We consider a set of training examples $\mathbb{X}$ where each example is denoted $x^{(i)}$ with $i \in [1...|\mathbb{X}|]$ and associated with a label $y^{(i)}$. The set of training examples $\mathbb{X}$ is composed of a set of features $j \in \mathbb{J}$ and each feature vector is noted x_j with $j \in \mathbb{J} = [1...N]$.

A *concept* is a subset of features denoted $c_k \subset \mathbb{J}$. K concepts c_k co-exist to form the set of concepts $c_k \in \mathbb{C}, k \in [1...K]$. The instantiation of a concept c_k is the process of populating it with dependent features. Every feature $j \in \mathbb{J}$ belongs to a single concept c_k and one concept only:

$$c_k \cap c_l = \emptyset \mid \forall l \in \mathbb{J} \text{ and } l \neq k$$

2.1. Expert knowledge concepts

The instantiation of a concept c_k can be either driven by domain knowledge or performed automatically. The former requires that all variables belong to user-defined groups that be meaningful to domain experts. The variable classifications are sometimes to be found in the documentation of a dataset. That is the case of the FRED-MD data, which is used in the experimentation section of this work. The paper accompanying the dataset (McCracken & Ng, 2016) includes in appendix a table that classifies the 134 monthly

macroeconomic indicators into 8 categories: output and income, labour market, housing, consumption orders and inventories, money and credit, interest and exchange rates, prices, and stock markets. Table 1 provides a sample of these categories.

2.2. Automatic concepts: a simple approach

Failing that user may rely on domain knowledge, the set of *concepts* $\mathbb{C}$ can be built automatically using a clustering algorithm based on feature correlations. Features (indexed by j) can be grouped in a concept c_k using any dependence measure ρ. The most straightforward is the Pearson correlation, that measures linear correlation between variables. Assuming the measure has values between $[-1; 1]$ (an absolute value of 1 meaning two variables perfectly dependent), a user-defined threshold ϵ is set on the absolute value of the measure of dependence between two features x_j and $x_{j'}$ in order to decide whether these features belong to the same concept c_k:

$$|\rho(x_j, x_{j'})| \geq \epsilon \mid \forall (j, j') \in c_k$$

The clustering algorithm is greedy: for each iteration a feature is tested against all features and existing groups. A feature j' is affected to a *concept* c_k if its dependence to each feature in c_k is higher than ϵ:

$$|\rho(x_j, x_{j'})| \geq \epsilon \mid \forall j \in c_k \rightarrow c_k = c_k \cup j'$$

If a given feature is independent from all the others, it belongs to a singleton. This formalization is also adequate for the expert's knowledge grouping. In that case, ρ and ϵ would be the criteria of group assignment by the expert.

The next section explains how the notion of *concepts* may be used to constrain the training of a decision tree in order to produce more interpretable surrogates.

3. Concept Tree: Embedding Concepts For More Interpretable Surrogate Decision Tree

Decision trees are a well-known interpretable machine learning model. A decision tree has a graphical structure, its decisions rely on a sparse subset of features, and features are used in a hierarchical way, thus conveying an intuitive sense of feature importance and providing several levels of explanation granularity (Freitas, 2014). Training a decision tree on the training set $\mathbb{X}$ yields an interpretable classification algorithm, provided that the number of nodes is kept under a certain threshold. The limit on the tree complexity may come at the expense of predictive performance. Decision trees appear as good candidate surrogates to black-box classifiers.

A decision tree surrogate is produced as follows. A black-box b is trained on $\mathbb{X}$ with the true class labels $y^{(i)}$; the surrogate f is then trained on the black-box predictions $\hat{y}^{(i)} = b(x^{(i)})$. In production, the classification is performed by the black-box while the explanations are provided by the surrogate decision tree. The fidelity of the surrogate is assessed as the proportion of instances where the surrogate makes the same prediction than the black-box classifier.

The TREPAN algorithm is an instance of interpretable surrogate tree model (Craven & Shavlik, 1996). It is model-agnostic and aims at mimicking the classification behaviour of a black-box b. It queries the black-box with instances to get predictions $\hat{y}^{(i)} = b(x^{(i)})$ and then fits an interpretable decision tree. The outline of TREPAN is shown in Algorithm 1. The querying of extra instances allows to populate the critical areas of the feature space and thus significantly curb the tendency of decision trees to overfit.

TREPAN uses $m - of - n$ decision rules, that are inspired from $id2 - of3$ decision trees (Murphy & Pazzani, 1991). To fit an $m - of - n$ decision rule, the set of the n most discriminative tests on the features for the node is discovered using the information gain. Then, in order to validate a node, an instance must validate at least m tests among the n. For instance, given a decision rule with 3 tests x_1, x_2 and x_3, such as 2-of-$\{x_1, \neg x_2, x_3\}$ is equivalent to the logical expression $(x_1 \vee \neg x_2) \wedge (x_1 \vee x_3) \wedge (\neg x_2 \vee x_3)$. The parameters m and n are user-defined upper-bounds: their final values are learnt by the node. The $m - of - n$ decision rules are learnt in a greedy way for computational efficiency. For the outline of the fitting algorithm of an $m - of - n$ decision rule, we refer the reader to the original paper (Craven, 1996) for the sake of conciseness and precision.

While the original TREPAN paper is two decades old already, researchers have kept reassessing its relevance up until recently (Sarkar et al., 2016). Experimentations show that TREPAN has a good fidelity to the black-box and a better accuracy on the test set than a decision tree directly trained on the training set $\mathbb{X}$ (Craven & Shavlik, 1996). This good performance is attributed to the additional-instance-drawing mechanism, which yields a denser support to the fit of a decision rule and thus a better prediction accuracy.

The $m - of - n$ decision rule structure improves the accuracy and the fidelity of the decision tree as it allows to learn more complex decision boundaries. However, it comes at the price of interpretability of both the node's decision rule and the decision tree overall. A practitioner may find it hard to understand all the possible $\binom{n}{m}$ combinations of variables at the same time. Moreover, the contrary of a $m - of - n$ literal may be challenging to conceive as soon as $m > 1$ and $1 < n < m$. Alternatively, simpler nodes such as the ones in $C4.5$ (Quinlan, 1993) would be easier to interpret, but would yield significantly larger trees for

Table 1. Overview of the grouping of variables by concept in FRED-MD database (McCracken & Ng, 2016)

Concept 1: Output and Income	Concept 2: Labor Market	Concept 5: Money and Credit
Real Personal Income	Civilian Labor Force	Total Reserves of Depository Institutions
Real personal income ex transfer receipts	Civilian Employment	Commercial and Industrial Loans
IP: Consumer Goods	Civilian Unemployment Rate	Total Consumer Loans and Leases Outstanding
...	...	...

a given fidelity level, thus reducing overall interpretability (Craven & Shavlik, 1996).

Similar to TREPAN, the Concept Tree aims at mimicking the classification behaviour of b, using the additional-instance-drawing procedure at the node level to improve fidelity. The chief difference between Concept Tree and TREPAN lies in the learning of the decision rule at each node. The $m-of-n$ decision rule is no longer fitted on the whole feature space. It is constrained to use subsets of features defined by the *concepts*. At a given node, the Concept Tree fits a $m-of-n$ decision rule using only the variables related to the *concept* c_k for each *concept* c_k, and selects the one that yields the best information gain. Each node thus splits the sample on a $m-of-n$ decision rule based on a *concept*, using related variables only. A part from that restriction, the Concept Tree uses the same expanding procedure as TREPAN, described in Algorithm 1.

This paper aims at improving the interpretability of surrogate trees built with $m-of-n$ decision rules by introducing the Concept Tree, a tree-based surrogate methods based on TREPAN and the use of *concepts*. The use of *concepts* is expected to help practitioners better understand the surrogate. Each node relies on variables belonging to one *concept*-grouping only. Nodes thus use complex $m-of-n$ literals but ensure better human-understandability by organizing information at a *concept* level.

We argue that concept-based decision rules have a better interpretability than standard $m-of-n$ decision rules while having the exact same informational complexity (the number of bits needed to write down the decision rule). Defying the conventional notion of complexity-interpretablity trade-off, Concept Trees achieve a higher interpretability at the same level of complexity, thus preserving predictive accuracy.

Algorithm 1 Simplified overview of Trepan

Trepan(b, X, max_nodes, min_sample, m, n)
Initialize the tree with root R
$S \leftarrow X \cup DrawSample(min_sample - |X|)$
Get labels from black-box b for train set S
Initialize $Queue$ with $< R, S >$
$n_nodes = 1$
while $Queue \neq \emptyset$ and $n_nodes < max_nodes$ **do**
  Remove $< nodeN, S_N >$ from head of $Queue$
  Fit decision rule of node N
  for each outcome t of the test **do**
    Initialize a child node C
    $S_c \leftarrow$ instances of S_N with outcome t for the test
    $S_C \leftarrow S_c \cup DrawSample(min_sample - |S_c|)$
    Get labels from black-box b for S_C
    if C is not pure enough **then**
      Add $< nodeC, S_C >$ to $Queue$
    end if
    $n_nodes = n_nodes + 1$
  end for
end while
Return R

4. Experimentation: FRED-MD Macroeconomic Database

This paper has introduced the ideas of Concept and Concept Tree, whose main objectives are to provide an accurate surrogate f mimicking a black-box classifier b while being as interpretable as possible. The next paragraphs describe experimentations made with the FRED-MD dataset (McCracken & Ng, 2016), a public macroeconomic database of 134 monthly U.S. indicators and more than 700 instances released by the Federal Reserve Bank of St. Louis. Interpretability is critical in economics and our experimentations show how Concept Trees may match the requirements of the field.

The experimentations applied to a downturn prediction exercise. Downturns are defined using the NBER Business Cycle Chronology, that provides dates for each peak and through in the US business cycle. The target variable is defined as "a downturn occurs in the next 12 months". Formally, the target variable y_t equals 1 if a downturn occurs between t and $t+12$, and 0 otherwise, as in (Wright, 2006). Downturn are predicted using the wide array of macroeconomic variables in FRED-MD. Domain-knowledge-based experts are extracted from the FRED-MD official documentation, which classifies variables into 8 subgroups (see Table 1).

The competitors are both flavors of Concept Tree (Concept Tree-Expert and Concept Tree-Correlated for automatically-

Algorithm 2 Construction of a Concept Tree decision rule

ConstructConceptDecisionRule(X, y, $concepts$)
$best_candidate \leftarrow \emptyset$
$best_ig \leftarrow= 0$
for $c \in concepts$ **do**
  $X_c \leftarrow$ Select features from X belonging to c
  $candidate \leftarrow MofNDecisionRule(X_c, y, m, n)$
  $ig \leftarrow$ Compute information gain for $candidate$
  if $ig > best_ig$ **then**
    $best_ig \leftarrow ig$
    $best_candidate \leftarrow candidate$
  end if
end for
Return $best_candidate$

defined concepts) and the original TREPAN. Since the Concept Tree and TREPAN have a similar structure, they share the same parameters for the experimentation. The maximal number of nodes max_nodes is set to 10. For the split rules, the values of $m-of-n$ are set to $1-of-1$, $3-of-3$ and $5-of-5$. The minimal value of samples $min_samples$ to fit a split rule at a node is 100, thus additional samples are drawn from the fitted distribution if the $\mathbb{X}$ is not large enough. For Concept Tree-Correlation, the threshold ϵ on the correlation ρ is set to 0.9 such as $|\rho(x_j, x_{j'})| \geq 0.9$.

The black-box b used is a Gradient Boosting Trees with 500 estimators. The learning rate and tree depth are grid-searched while other parameters have standard scikit-learn values (Pedregosa et al., 2011). Out-sample-fidelity is computed by 50-split time-series cross-validation. Time-series cross validation implies that the train set includes all observations prior to t, for corresponding to January 1st of each of the last 50 years in the sample. The test set includes the 12 months following t. At each split the black-box is fitted on the train set and makes predictions for the train set and the test set. The Concept Tree and TREPAN instances are then fitted on the train set with black-box predictions as targets, and their fidelities are measured against the black-box predictions made on the test set. Out-of-sample accuracy is assessed using the same procedure. Fidelity measures the proportion of predictions made by the surrogate that match the predictions made by the black-box, while accuracy measures the proportion of predictions made by the surrogate that match the actual value of the target. Interpretability is assessed by economic expert judgement.

4.1. Results

Table 2 presents the cross-validated accuracies and fidelities for TREPAN, the Concept Tree with expert-defined concepts and the Concept Tree with automatically defined clusters. The black-box mean accuracy over the folds is $84\% \pm 27\%$.

Table 2. Experimental results: surrogate accuracy and fidelity as a function of the algorithm, the concept type and the split rule

Algorithm	Concept Type	Split Rule	Surr. Accuracy	Surr. Fidelity
Concept Tree	Expert		$83\% \pm 29\%$	$85\% \pm 27\%$
Concept Tree	Correlation	$1-of-1$	$82\% \pm 30\%$	$85\% \pm 28\%$
TREPAN	/		$84\% \pm 29\%$	$86\% \pm 26\%$
Concept Tree	Expert		$78\% \pm 32\%$	$83\% \pm 28\%$
Concept Tree	Correlation	$3-of-3$	$83\% \pm 27\%$	$84\% \pm 25\%$
TREPAN	/		$84\% \pm 29\%$	$89\% \pm 22\%$
Concept Tree	Expert		$77\% \pm 31\%$	$82\% \pm 26\%$
Concept Tree	Correlation	$5-of-5$	$84\% \pm 25\%$	$82\% \pm 25\%$
TREPAN	/		$82\% \pm 30\%$	$86\% \pm 22\%$

The experimentations show the relevance of this family of interpretability approaches in terms of accuracy and fidelity to the predictive model. The experimentations also show that Concept Tree provides surrogates whose fidelity and accuracy matches the performance of TREPAN trees and whose interpretability may be significantly enhanced. Although, TREPAN overall slightly leads in terms of accuracy and fidelity, the non-negligible standard-deviations and the setup of this preliminary experiment (number of folds and datasets) don't allow for a final conclusion. However, the experiment highlights the relevance of Concept Tree in terms of accuracy and fidelity and as things stand, Concept Tree is at least as relevant as TREPAN.

Figure 2 in Appendix displays the trees generated by TREPAN (Figure 2(a)), the Concept Tree with expert-defined concepts (Figure 2(b)) and the Concept Tree with correlation-based defined concepts (Figure 2(c)).

In Concept Tree-Expert (based on domain-knowledge), explanations are structured by sharing a common "language" with users or experts. It provides the big picture with one general *concept* by node. The detailed analysis of a node is eased because only related, homogeneous, variables are assembled. There is an intuitive relations between high-level explanations (concepts) and low-level explanations (raw variables).

In Concept Tree-Correlation, computed automatically based on variable correlations, part of the domain-knowledge can be recovered. Concept Tree-Correlation presents also the advantage of gathering dependent variables for each node, avoiding arbitrary choices between correlated variables to build a test.

In contrast, TREPAN trees use an idiosyncratic language not shared by the practitioner. Associations of tests in a TREPAN node generate confusion by gathering variables that are hardly related from a domain-knowledge point of view. Such nodes obstruct the understanding by preventing the user from getting the big picture.

From a macroeconomic point of view, the Concept Tree-Expert (2(b)) yields meaningful high level explanations of

the workings of the black-box classifiers. The cascade structure of the tree indicates that the black-box algorithm would only predict a downturn when five conditions occur simultaneously, with regard to interest rates, unemployment, housing markets, prices and household income. The first condition corresponds to high spread for either public or private bonds, which both indicate high levels of uncertainty, as lenders may ask for higher risk premia when facing higher downturn likeliness. The second condition implies that if unemployment is already high or increasing, the economy is already in recession, which makes the occurrence of a downturn unlikely. Two conditions within the third node apply to the same variable: a number of housing permits in the West lower than 255 or higher than 558 along with an increasing number of permits in the Midwest is captured as a pattern conducive to downturns. The dual threshold on permits in the West fits with economic intuition, whereby too low housing construction may indicate low growth and too high construction can hint at overheating of the housing market or bubbles. The three tests in the price concept node correspond to a price index growing at a pace lower than 1%, either the Consumer Price Index or its durable goods subcomponent, or the Producer Price Index. Overall, Concept Tree enhances the interpretability of surrogate trees by **structuring** the explanations.

5. Conclusion

The present paper introduces *concepts*, a meaningful manner to group dependent variables, and Concept Trees, an alternative tree-based surrogate model that provides both high-level and detailed explanation to black-box classifiers. The grouping of variables in *concepts* allows to overcome the false sense of simplicity conveyed by simpler decision tree surrogate that may give an artificially high importance to a given variable picked among a set of correlated variables, thus obscuring the bigger picture. The use of *concepts* also helps practitioners make sense of otherwise cryptic $m - of - n$ literals, by relying on a higher-level representation of the data. Compared to TREPAN, Concept Trees produce surrogates that have a comparable size and are as accurate, but more easily understandable to a human thanks to a better organization of the information along higher-level representations that significantly enhance the interpretability of the surrogate.

Experiments were conduced using FRED-MD, a macroeconomic database whose documentation includes a grouping of variables. The Concept Tree was applied to this data using both expert-defined concepts derived from the data documentations and concepts built using a simple correlation-based clustering algorithm. First results show a notable improvement in human-readability while accuracy and fidelity of the surrogate are preserved.

Further research could involve a deeper assessment of our propositions, both quantitatively and qualitatively. It could also be relevant to explore alternative clustering algorithms designed to produce more relevant *concepts*. Further modification to the Concept Tree algorithm may improve performance: currently, following TREPAN's principle, one concept can only be used once in a decision path. Considering a concept encompass several variables, the accuracy and fidelity of the surrogate may suffer from this probably too severe constraint.

References

Bühlmann, P., Rütimann, P., van de Geer, S., and Zhang, C. H. Correlated variables in regression: Clustering and sparse estimation. *Journal of Statistical Planning and Inference*, 143(11):1835–1858, 2013.

Craven, M. Extracting comprehensible models from trained neural networks. *PhD Thesis*, 1996.

Craven, M. W. and Shavlik, J. W. Extracting Tree-Structured Representations of Trained Networks. *Advances in neural information processing systems*, pp. 24–30, 1996.

Freitas, A. A. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014.

Ghorbani, A., Wexler, J., and Kim, B. Automating Interpretability: Discovering and Testing Visual Concepts Learned by Neural Networks. 2019.

Gregorutti, B., Michel, B., and Saint-Pierre, P. Correlation and variable importance in random forests. *Statistics and Computing*, 27(3):659–678, 2017.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Pedreschi, D., and Giannotti, F. A Survey Of Methods For Explaining Black Box Models. *ACM Computing Surveys*, 51(5), 2018.

Kim, B., Patel, K., Rostamizadeh, A., and Shah, J. Scalable and interpretable data representation for high-dimensional, complex data. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). 2017.

McCracken, M. W. and Ng, S. Fred-md: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589, 2016.

Murphy, P. M. and Pazzani, M. J. Id2-of-3: Constructive induction of m-of-n concepts for discriminators in decision trees. In *Machine Learning Proceedings 1991*, pp. 183–187. Elsevier, 1991.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Quinlan, J. R. C4. 5: Programs for machine learning. 1993.

Sarkar, S., Dragicevic, S., and Percy, C. Accuracy and Interpretability Trade-offs in Machine Learning Applied to Safer Gambling. In *CEUR Workshop Proceedings*, 2016.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

Strobl, C., Boulesteix, A.-l., Kneib, T., Augustin, T., and Zeileis, A. Conditional variable importance for random forests. *BMC Bioinformatics*, 9(307):1–11, 2008.

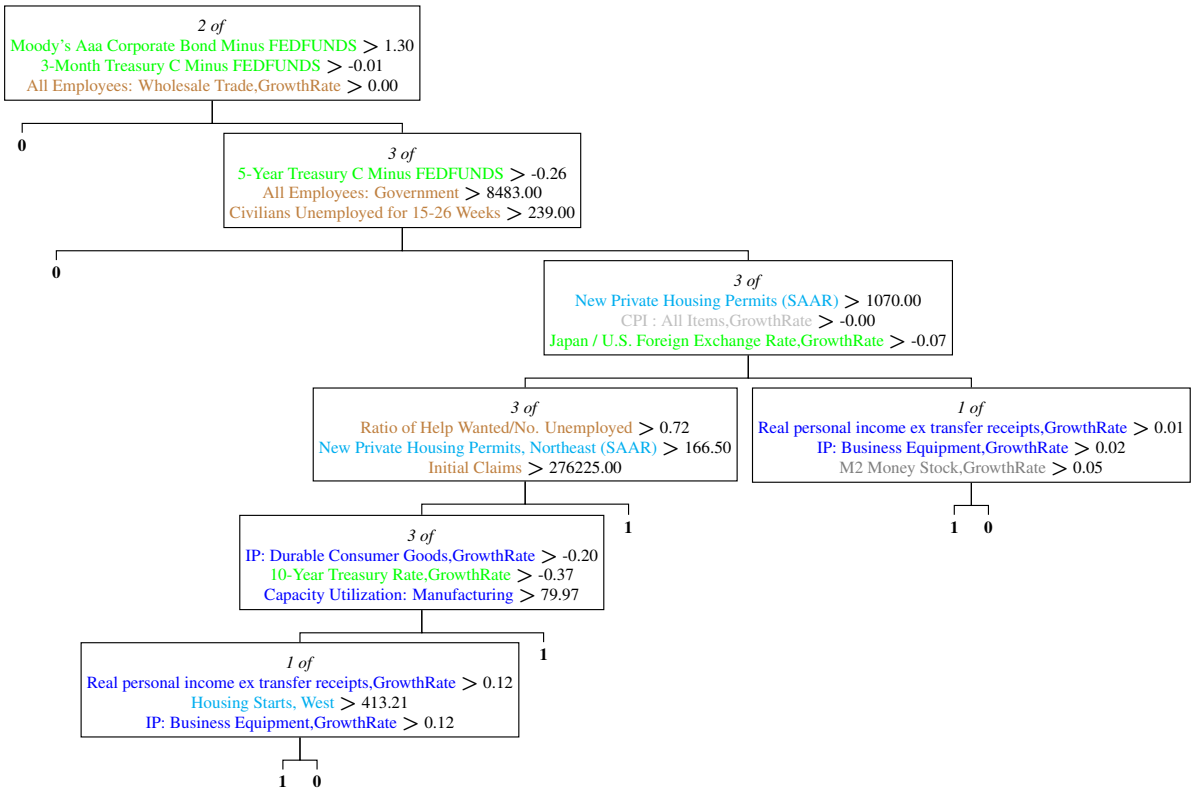
Tolosi, L. and Lengauer, T. Classification with correlated features : unreliability of feature ranking and solutions. *Bioinformatics*, 27(14):1986–1994, 2011.

Vidal, R. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011.

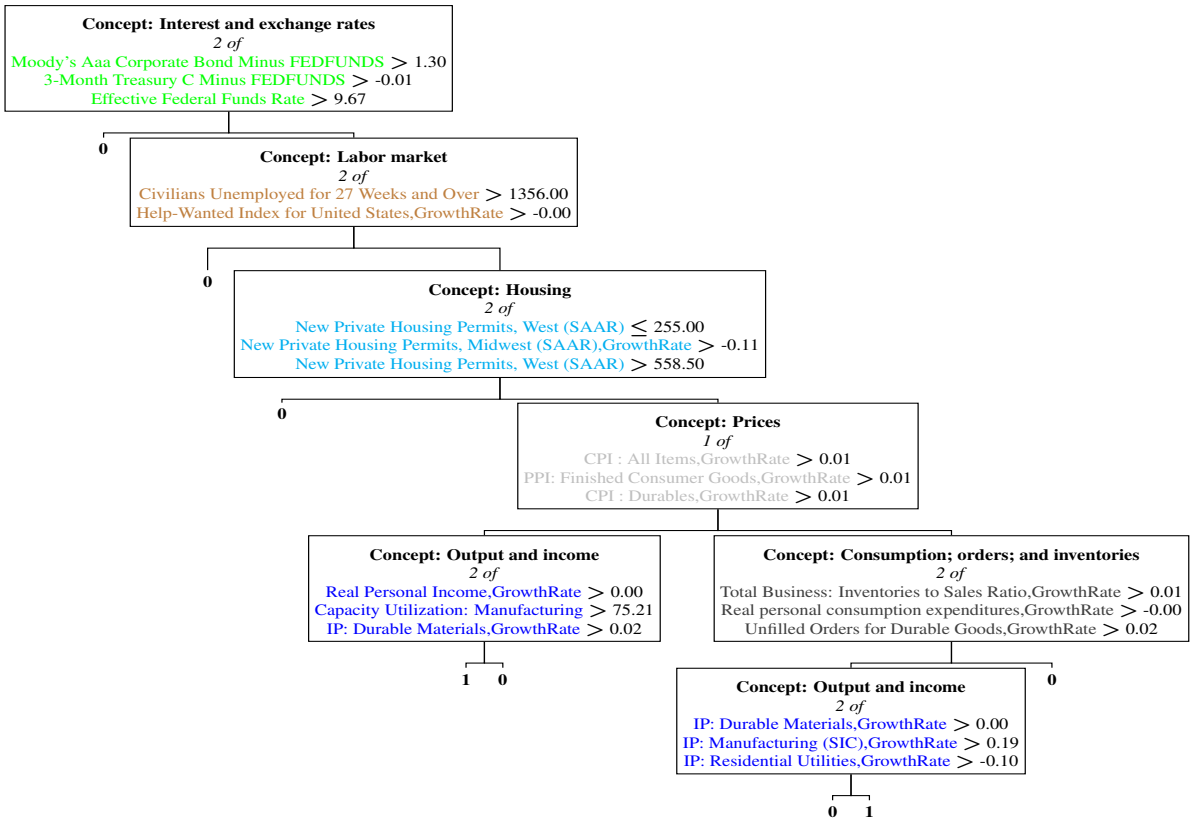
Wright, J. H. The yield curve and predicting recessions. *FEDs Working Papers*, 7, 2006.

Yuan, M. and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68 (1):49–67, 2006.

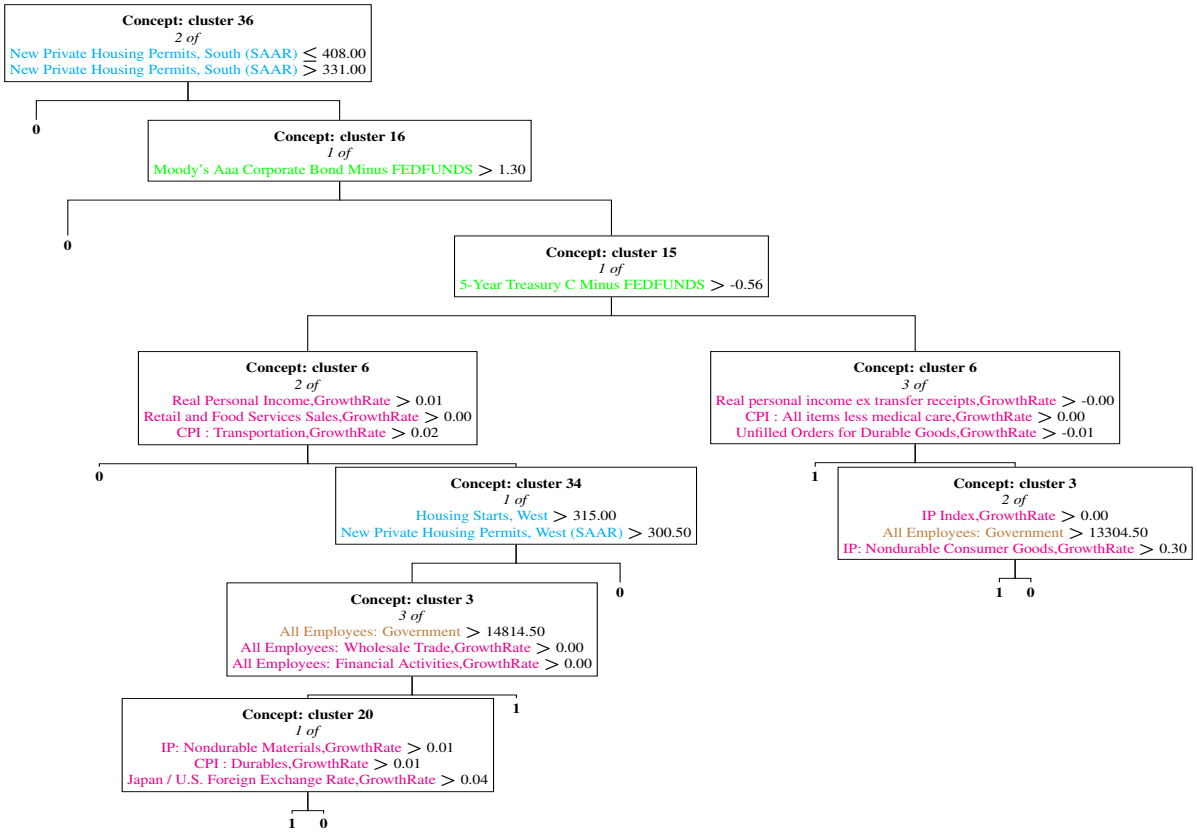
A. Appendix: Decision Trees



(a) **Trepan Decision Tree. Variables from the same expert-defined concepts are displayed with the same color. We can thus easily see that most nodes use variables from heterogeneous groups, making the interpretation difficult**



(b) **Concept Tree-Expert. Variables from the same expert-defined concepts are displayed with the same color. Each node use variables from one single expert-based concept**



(c) **Concept Tree-Correlated. Each node use variables from one single concept based on variable correlations. Variables from the same expert-defined concepts are displayed with the same color.**

Figure 2. **Structure of trained** $3-of-3$ **Decision Trees**