## 1.data science

#product information (title, description, price, …) manually associated categories.

#multiclass classification problem

#decide the confidence threshold required for a product to be categorized automatically

#accuracy-coverage

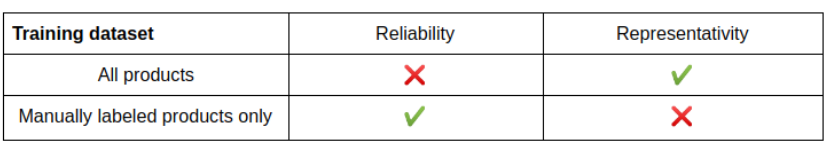
coverage is the proportion of categorized products, i.e. products which probability is higher than the threshold

accuracy is the proportion of correctly categorized products among the categorized ones (i.e. among those with probability higher than the threshold)

## 2. approach not sufficient

#The model deteriorates because it feeds on its past mistakes.

#one of our approaches satisfy reliable and representative of the whole flow



## 3. Human-in-the-loop to-the-rescue

#chose to sample randomly and every day some incoming products (categorized by the algorithm or not) that should be checked by hand by business owners to guarantee representativity and reliability.

# Django to build a labelling interface

#aimed at: clean (reliable and representative) dataset to train their algorithms

business teams to improve categorization quality by themselves and have a frequent feedback on their impact on accuracy

4.conclusion

Solution: sampling random products every day and have them labelled by humans.

**Flat Classification Ansatz**

Hierarchical Classification with Local Classifiers: Down the Rabbit Hole

## Approaches of designing a hierarchical classification model.

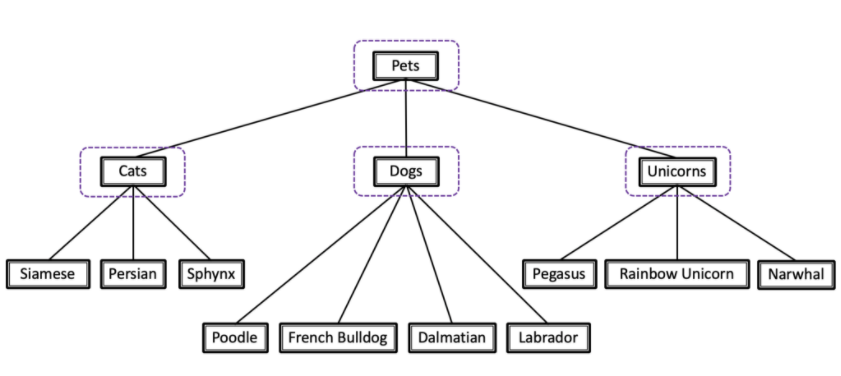
building an ensemble model comprised of hierarchically-structured local classifiers.

1. Local Classifier per Node (LCN)(binary classifier for each node)

+naturally multi-label :single-label algorithm , assign the label with highest confidence

-the inconsistency problem: the parent node votes no, but the current node vote yes.

-requires a lot classifiers

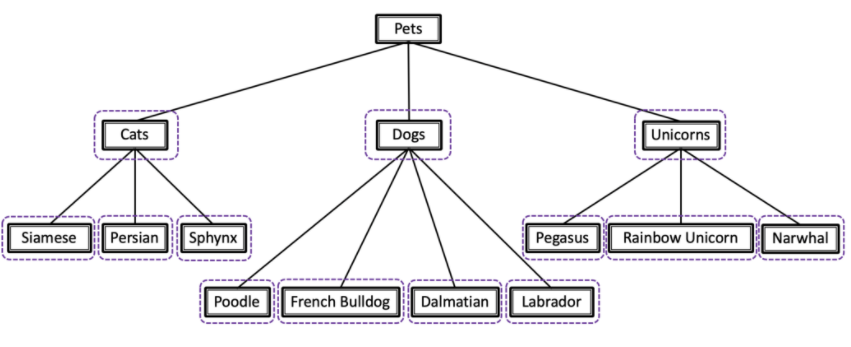


1. Local Classifier per Parent Node (LCPN)(each parent node gets one multi-class classifier)

+ structured in a way that others can easily grasp

+easier to implement

-inconsistency



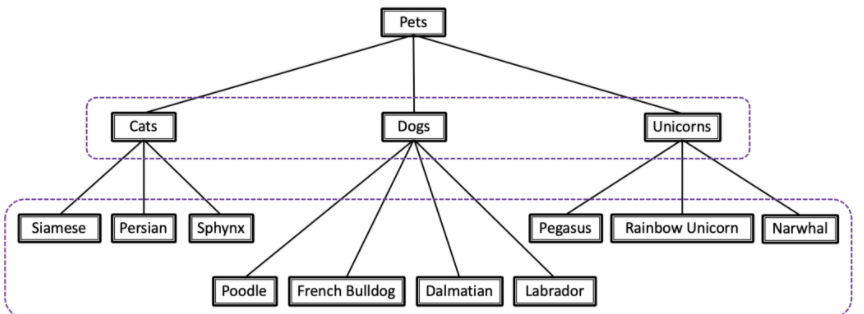
1. Local Classifier per Level (LCL) (multi-class classifier for each level of taxonomy, first classifier distinguish between parent second distinguish all of the leaf-node )

+very few classifier, few lines of code

-inconsistency

-at deep levels of the taxonomy classifier might have way too many classes to choose from.

-completely ignores parent-child relationships



## LCPN is the best

if you’ve got a multi-label problem, it’s the natural choice.

If it’s a single-label problem, though, I’d go with the LCPN approach. Its inconsistency problem is also very easily solvable — and that solution is pretty intuitive, too

**The Hitchhiker’s Guide to Hierarchical Classification**

## Problem:

human mind----------hierarchical structures

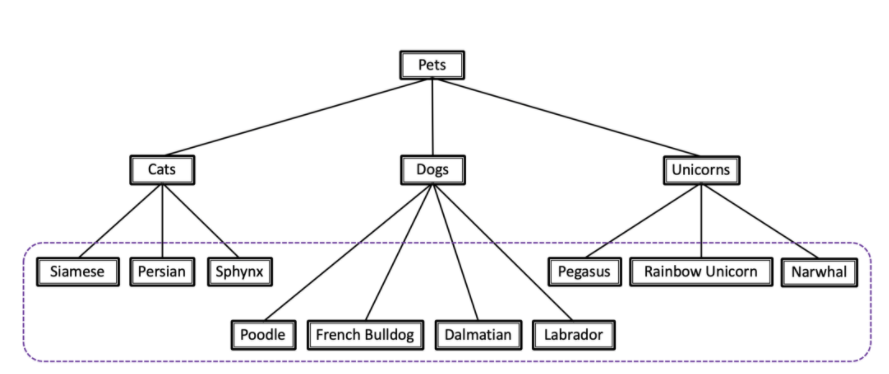
data science-----------receive input is flat

## Flat classification

 classify each example to its final, leaf-level label.

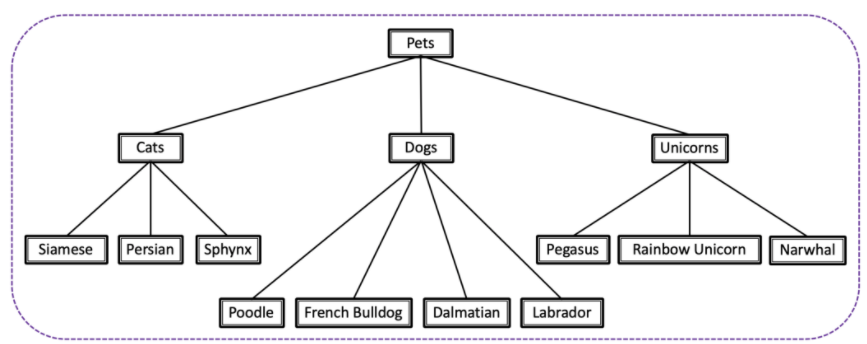
+simple

-lose important information(hierarchy infomation)



## The Big Bang Approach

Global classifier, considers the entire class hierarchy as a whole, during a single run.



## Hierarchically-Structured Local Classifiers

LCPN/LCN/LCL

**Hierarchical Classification by Local Classifiers: Your Must-Know Tweaks & Tricks**

## Inconsistency Correction Method

1. Predictions at Different Levels Contradict Each Other

Solution: class-prediction top-down approach

e.g. During the testing phase, you start from the top of the taxonomy tree, and every time you go down a level, you only consider the local classifiers that are the descendants (后代)of the predicted class

1. Several Leaf-Node Labels for a Single-Label Problem (get multiple local classifiers in the same level saying “yes” on the same example, here’s what you do: go with the highest-confidence one. )

Solution: top-down approach, sending examples to child-classifiers of parent-classifiers that came out positive

## Error Propagation

Solution: the confidence score for a prediction doesn’t hit it — stop the classification process there, and do not send that sample on to that classifier’s child-nodes.

Problem of the solution: he final prediction would be less specific

## Positive & Negative Examples for LCN Classifiers(Binary classifier)

1. ”the siblings policy” the most sense for LCN

Positive for class X: class X and its child

Negative for class X: class X’s siblings-class and its child-class

## Same vs. Different Features/Algorithm

Could not use the same base algorithm for all of the local classifiers