

PLONE 5 & ML

THE MOST POWERFUL PYTHON OPEN
SOURCE CMS

Created by Ramon Navarro Bosch / @bloodbare

RAMON NAVARRO BOSCH

- Plone Foundation Member
- Plone Framework Team
- CTO at Iskra.cat
- Catalan



THE TALK

- PLONE 5
- Machine Learning

PLONE 5

- Quality
- Tests
- Protect investment in current sites
- Usability for real people
- Stability
- Scale from S to XL

A large, semi-transparent watermark of the PLONE logo is positioned in the background. The logo consists of a stylized blue 'P' icon followed by the word 'PLONE' in a bold, sans-serif font, with a registered trademark symbol (®) at the end.

WHO DOES NOT KNOW PLONE?

From the user point of view



CONTENT

Content is the king (Dexterity)

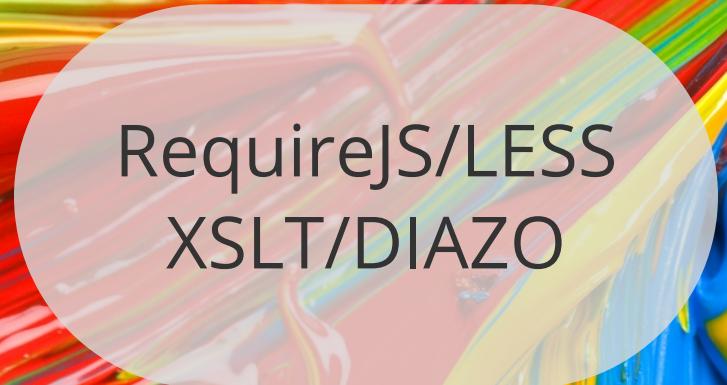
With full multilingual

GRANULAR SECURITY

Zope/ZODB
Workflow

The background of the slide features a vibrant, abstract composition of various colors of paint (red, yellow, blue, green) mixed together in a swirling, expressive manner, creating a dynamic and artistic feel.

THEMING

A large, semi-transparent white circle is positioned in the center of the slide, containing the following text:

RequireJS/LESS
XSLT/DIAZO

A screenshot of a Plone inline editor interface. The top bar shows standard browser controls (Back, Forward, Stop, Refresh) and a search bar with placeholder "Search Site". Below the search bar is a "Format" button and an "Insert" button. The main content area displays the Plone logo and a navigation menu with "Home", "News", "Events", and "Users". A breadcrumb trail indicates the current location is "Home". The main title "Plone Mosaic" is displayed. Below the title, a large heading "INLINE EDITION / TTW" is shown. To the left of the heading, there is a sidebar with the text "PLONE MOSAIC SPRINT BARCELONA". To the right of the heading, there is a text area containing a paragraph of Latin text. A blue rectangular box highlights a list of bullet points at the bottom of the text area. At the very bottom of the page, a footer bar contains links for "Site Map", "Accessibility", "Contact", and "POWERED BY PLONE & PYTHON".

localhost:8080/Plone/-/add/-/Document

Save Cancel Properties B I Format Insert

Plone

Search Site Search

only in current section

Home News Events Users

You are here: Home

Plone Mosaic

INLINE EDITION / TTW

PLONE MOSAIC SPRINT BARCELONA

Donec ultricies nisi ut felis, suspendisse potenti. Lorem ipsum ligula ut hendrerit mollis, ipsum erat vehicula risus, eu suscipit sem libero nec erat. Aliquam erat volutpat. Sed congue augue vitae neque. Nulla consetetur portitor pede. Fusce purus morbi tincidunt magna condimentum vel, placerat id blandit sit amet tortor.

- Nam molestie nec tortor
- Donec placerat leo sit amet velit
- Vestibulum id justo ut vitae massa

The Plone® Open Source CMS/WCM is developed by the Plone Foundation and friends. Distributed under the GNU GPL license.

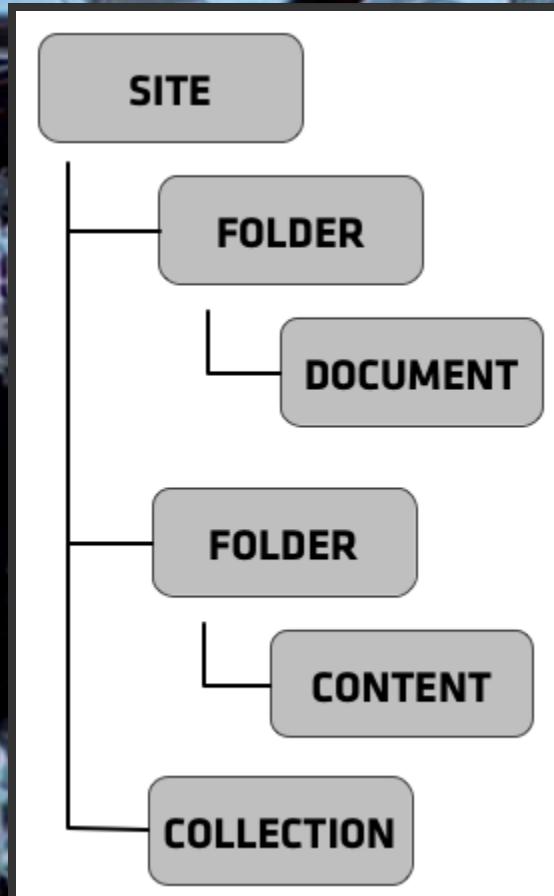
Site Map Accessibility Contact POWERED BY PLONE & PYTHON

TRY!

WHO DOES NOT KNOW PLONE

From the tech point of view

HIERARCHY



DEXTERITY

```
from plone.app.textfield import RichText
from plone.autoform import directives
from plone.namedfile import field as namedfile
from plone.supermodel.directives import fieldset
from plone.supermodel import model
from z3c.form.browser.radio import RadioFieldWidget
from zope import schema
from zope.schema.vocabulary import SimpleVocabulary
from zope.schema.vocabulary import SimpleTerm

from ploneconf.site import MessageFactory as _

LevelVocabulary = SimpleVocabulary(
    [SimpleTerm(value=u'platinum', title=_(u'Platinum Sponsor')),
     SimpleTerm(value=u'gold', title=_(u'Gold Sponsor')),
     SimpleTerm(value=u'silver', title=_(u'Silver Sponsor'))]
```

TRAVERSAL

`http://yoursite.cat/folder-i-created/my-document/my_view`

VIEWS

```
from Products.CMFCore.utils import getToolByName
from Products.Five.browser import BrowserView
from Products.CMFPPlone.resources import add_resource_on_request

class TalkListView(BrowserView):
    """ A list of talks
    """

    def __call__(self):
        add_resource_on_request(self.request, 'jquery.datatables')
        return super(TalkListView, self).__call__()

    def talks(self):
        results = []
        portal_catalog = getToolByName(self.context, 'portal_catalog')
        current_path = "/" . join(self.context.getPhysicalPath())
```

VIEWS II

```
<browser:page
    name="talklistview"
    for="*"
    layer=".interfaces.IPloneconfSiteLayer"
    class=".views.TalkListView"
    template="templates/talklistview.pt"
    permission="zope2.View"
    />
```

VIEWS III

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"
      metal:use-macro="context/main_template/macros/master"
      i18n:domain="ploneconf.site">
<body>

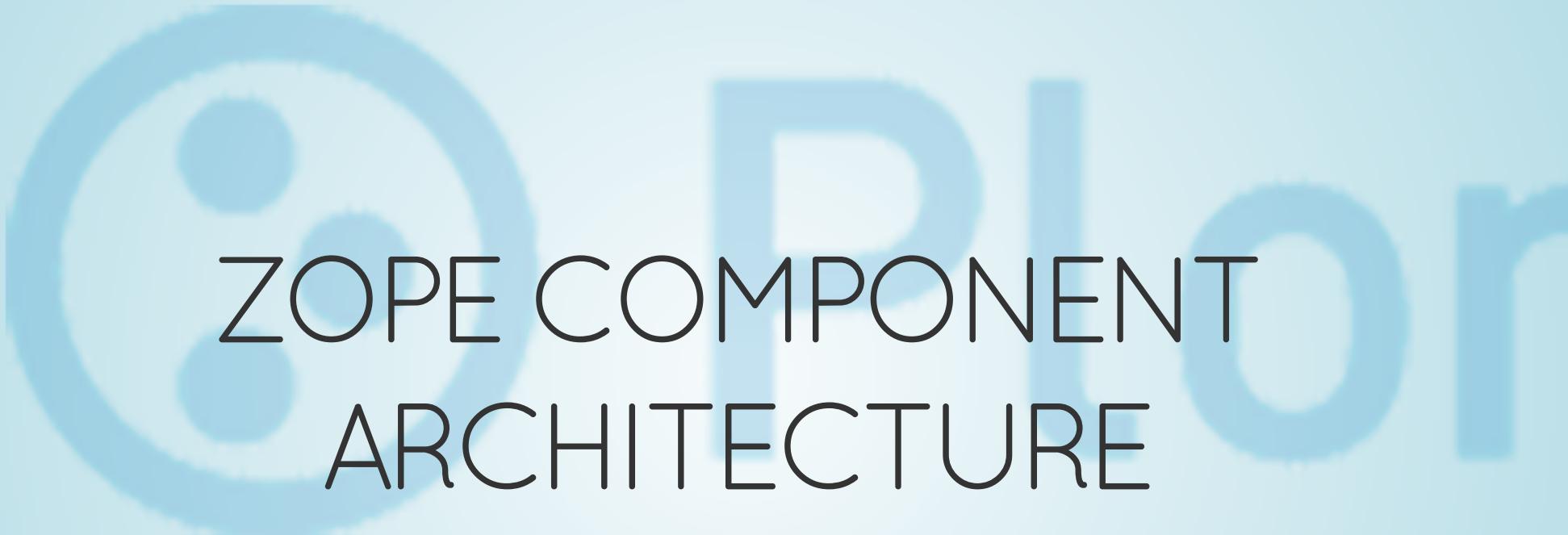
<metal:head fill-slot="javascript_head_slot">
    <script type="text/javascript">
        $(document).ready(function(){
            var oTable = $('#talks').dataTable({
            });
        })
    </script>
</metal:head>

<metal:content-core fill-slot="content-core">

    <table class="listing" id="talks">
```

PYTHONIC

```
site.folder.document.attribute
```



ZOPE COMPONENT ARCHITECTURE

STORAGE OF COMPONENTS

- global registry
- local registry

ADAPTER

```
class IPerson(Interface):  
  
    def whichTShirtIMWearing(self):  
        pass  
  
class CatalanGuy(object):  
  
    implements(IPerson)  
  
    def whichTShirtIMWearing(self):  
        return "estelada"  
  
class BaskGuy(object):  
  
    implements(IPerson)
```

SUBSCRIBER

```
<subscriber  
    for="object.IKindOfObject  
          zope.lifecycleevent.interfaces.IObjectModifiedEvent"  
    handler="function"  
/>
```

UTILITY

```
<utility  
    for=".interfaces.IGoodAlcohol"  
    handler=".object.GoodAlcohol"  
/>
```

JS/CSS

- Mockup/Patterns

```
<input class="pat-pickadate" type="text"
      name="form.widgets.IDublinCore.effective"
      value=""
      data-pat-pickadate="{
        'date': {
          'format': 'mmmm d, yyyy',
          'max': [2035, 1, 1],
          'min': [1915, 1, 1],
          'firstDay': 0,
          'selectYears': 200,
          'placeholder': 'Posa una data',
          'today': 'Avui'},
        'time': {
          'placeholder': 'Tria l\'hora',
          'today': 'Avui',
          'format': 'h:i a'}}>
```

- RequireJS/LESS

DIAZO

```
<?xml version="1.0" encoding="UTF-8"?>
<rules
    xmlns="http://namespaces.plone.org/diazo"
    xmlns:css="http://namespaces.plone.org/diazo/css"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <theme href="/++theme++abb.basictheme/index.html" css:if-not-content=>
        <theme href="/++theme++abb.basictheme/index-anon.html" css:if-content=>
            <notheme css:if-not-content="#visual-portal-wrapper" />

    <rules if-content="/*[@id='portal-top']">
        <!-- Attributes -->
        <copy attributes="*" theme="/html" content="/html"/>
        <!-- Base tag -->
        <before theme="/html/head/title" content="/html/head/base"/>
        <!-- Title -->
        <replace theme="/html/head/title" content="/html/head/title" />
```

REST API

plone.rest

```
<plone:service  
    method="PUT"  
    for="plone.dexterity.interfaces.IDexterityContent"  
    factory=".demo.Put"  
/>
```

```
class Put(Service):  
  
    def render(self):  
        return ISerializeToJSON(self.content)
```

TESTING

JENKINS.PLONE.ORG



... and the plone testing team

PLONE 5.0 BETA 3!

IS HERE!

```
git clone git@github.com/plone/jump.into.git  
python2.7 bootstrap.py  
  
# You need python-dev libpng-dev libjpeg-dev build-essential  
# libxml2-dev libxslt-dev (gcc)  
  
.bin/buildout -v  
  
# Coffee  
  
.bin/instance fg
```

TRAINING AND DOCS

- training.plone.org
- docs.plone.org

FUTURE (MY OPINION)?

- REST API
- asyncio
- Pyramid/SustanceD
- JS frontend

MACHINE LEARNING?

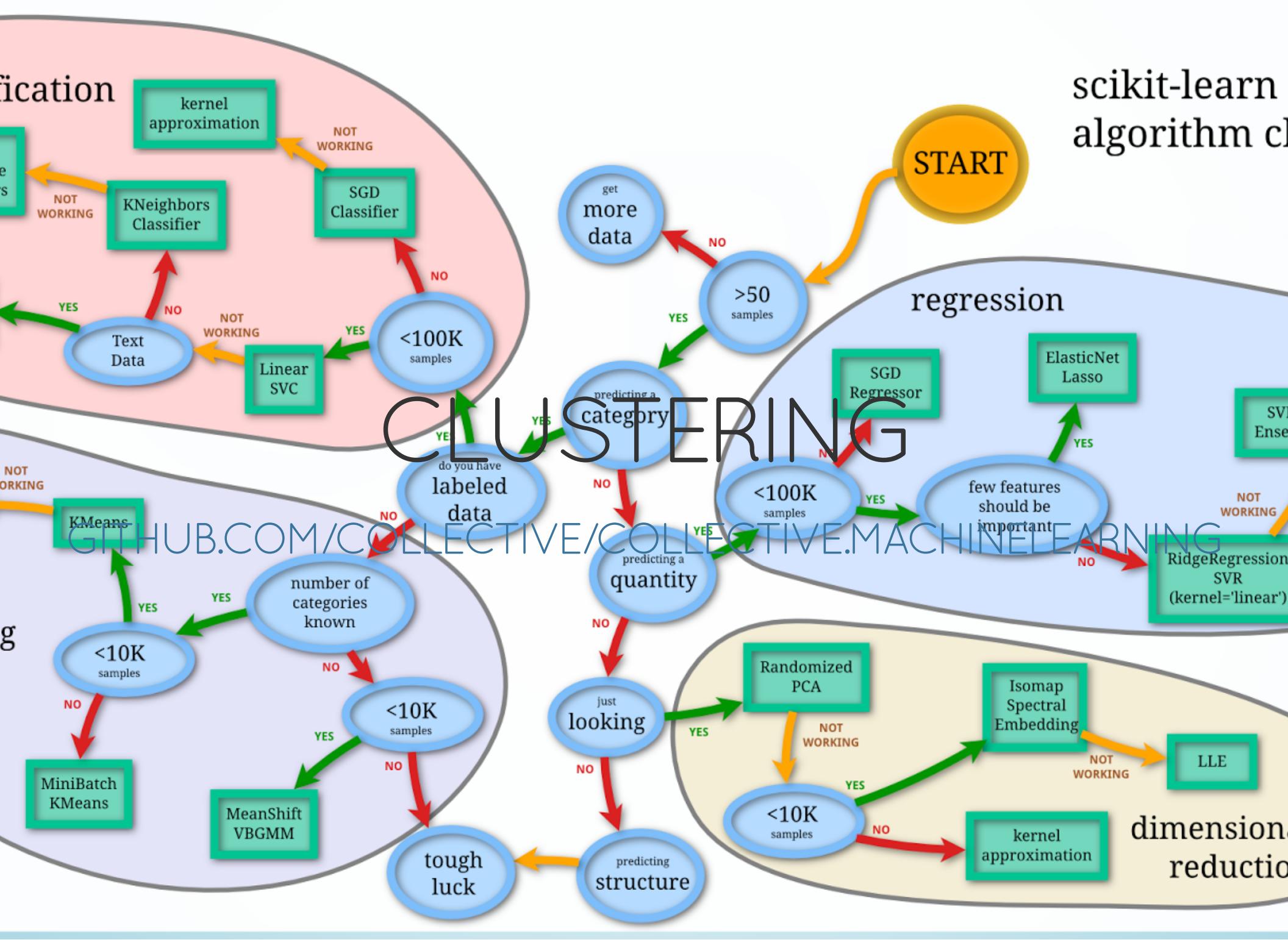
Content focused with sklearn



GOALS

- Related content
- Classification of current content with existing data set
- Recommendation of subject tags
- Semantic search on the live search

scikit-learn algorithm cheat sheet





ADAPTER ILEARNINGSTRING

```
from zope.component import getAdapter
from collective.machinelearning.interfaces import ILearningString

getAdapter(obj, ILearningString)
```

NORMALIZE AND VECTORIZE

- nltk.stem
- FeatureHasher

And we store the corpus as a pickle

CLUSTERIZE

- Define the number of clusters on FE
- KMeans!

And we store the model as a pickle

- We store on the objects to which cluster it belongs
- We can predict new objects to which cluster it belongs
- Security is implicit by Plone

FUTURE

- Classification (now trying Naive Bayes) on clusters (collective.gensim)
- Recommendation
- Semantic Search
- External and incremental computation

COMMUNITY



THE END

THANKS

@BLOODBARE

RAMON@PLONE.ORG