Article
Renato Banzai · Jul 17, 2020 3m read

Open Exchange

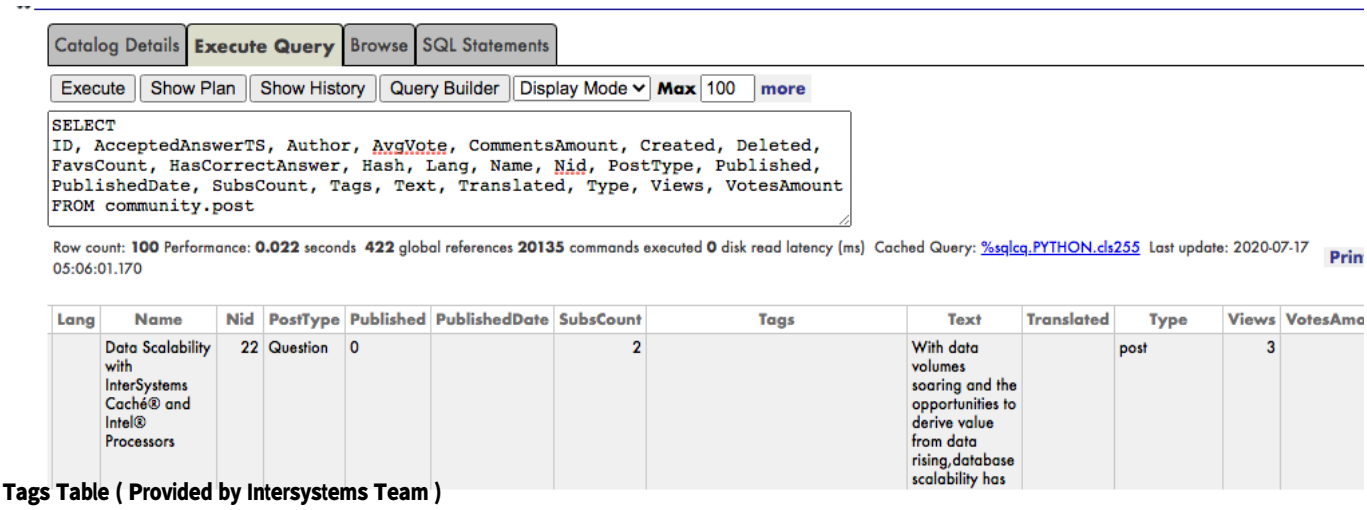
Using Machine Learning to Organize the Community - 2

This is the second post of a series explaining how to create an end-to-end Machine Learning system.

Exploring Data

The InterSystems IRIS already has what we need to explore the data: an SQL Engine! For people who used to explore data in
csv or text files this could help to accelerate this step. Basically we explore all the data to understand the intersection
(joins) which should help to create a dataset prepared to be used by a machine learning algorithm.

Posts Table (Provided by Intersystems Team)



Tags Table (Provided by Intersystems Team)

Our challenge demands to classify a **post** with the right **tags**. And we have a bunch of posts and all the tags with some description.
There are several classification techniques but the common cases of classification uses structured data! An article, a long text isn't exactly structured in this point of view.

Machine Learning works most of time with NUMBERS - Deal with it

Yes! Even the most normalized Data Model has **text** but when we have to use ML Algorithms we need to find a solution
to convert all texts into numbers but remember **unstructured data* will not take the numbers ready to be used in a classification model. It s time to...

Feature Engineering

In this step we convert the text, numbers, unstructured and chaos things in (most of time) a **matrix**... yes that one element of your past Algebra Classes. If we have a table that already looks like a matrix, probably you will have less
work than us! We have a big text with different sizes and shapes and words...

Bag of Words

One of the ways to transform a long text into a matrix is a Bag of Words:
expression
Nicole should buy a car
Jack should sell his boat

```
 [Nicole, Jack, should, buy, sell, a, his, car, boat]
 [1     ,    0,      1,   1,    0, 1,   0,   1,    0]
 [0     ,    1,      1,   0,    1, 0,   1,   0,    1]
```

Its just a small example. But you can see: the more there are words, the bigger the matrix. Fortunatelly we have a lot of
components to help you create Bag of Words. In this case I have used sklearn components to do this.

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.preprocessing import MultiLabelBinarizer

ml_bin = MultiLabelBinarizer(classes=all_tags)
matrix_y = ml_bin.fit_transform(a_dataframe)

count_vec = CountVectorizer(ngram_range=(1,1), tokenizer=tokenize, strip_accents='uni
code', stop_words=stop_words)
matrix_x = count_vec.fit_transform(a_dataframe)
```

After run the above methods we have a vectorizer and we will need to keep this vectorizer all the time we want to predict.
If we change the vectorizer we have the chance to mess up the matrixes and nothing should work after this.

I have already told you that we need to use only numbers?

There is another problem! For some algorithms the size of number can mess up your intentions. Briefly if one matrix element
has a val 1 and another element has value 987, some algorithms could interpret this as importance of the element and
take the wrong way.

Classification ML Algorithms

There is a bunch of material to read about classification algorithms, you can start with this article:
https://machinelearningmastery.com/types-of-classification-in-machine-learning/

Next article: training strategy (of data forget about gims)

I hope you are enjoying. If you like the text and my application, please vote on
https://openexchange.intersystems.com/contest/current in my application **iris-ml-suite**

#IntegratedML #Machine Learning (ML) #Python #Unstructured Data #Vector Search #InterSystems IRIS
Check the related application on InterSystems Open Exchange

Source URL:https://community.intersystems.com/post/using-machine-learning-organize-community-2