Classification of scanner data into COICOP: a machine learning approach.

Ongoing work

Adrien Montbroussous Martin Monziols

Table of contents

Abstract **1**

Introduction **2**

Data **2**
Scanner data description . . . 2
Product dictionary . . . 3
Data cleaning process . . . 4
Exclusions . . . 4
Variable cleaning . . . 5

Models and Methodology **5**
Classification model . . . 5
Methodology . . . 8
Performance on already classified, the "test sample" of the labelled dataset . . . 9
Sampling for manual classification and confidence interval calculation . . . 9
Manual classification in practice . . . 13

Results **14**
Performance on already labelled data (test sample) . . . 14
COICOP 6 digit level ("Postes") . . . 14
Performance on "unlabeled" data . . . 16
COICOP 6 digit level ("Postes") . . . 16
COICOP 5 digit level (Sub-Classes) . . . 17
COICOP 4 digit level (Classes) . . . 17
COICOP 2 digit level . . . 18
Analysis of the predictions at the COICOP 2 digit level . . . 19

Conclusions **23**
About the results 23
About our prediction strategy 23
About manual labelisation strategy 23
References 24

Abstract

Scanner data has been used in production to compute the HICP[1] and CPI[2] for France since January 2020, for most French retailers. Our current methodology with this data uses a product dictionary bought from an external provider containing detailed characteristics for each product. Importantly, these characteristics allow us to match articles in our sales data with the COICOP[3]. We recently obtained scanner data from two major hard discount retailers. However, our product dictionary currently does not cover their products, so making use of these data is far from being automatic. To make possible the production of an index with these new data, the first step is to be able to match the EAN[4] to the COICOP at a detailed enough level. To do so, we experiment with automatic classification methods based on machine learning models. Following our current methodology, we can easily build train data for our models: on the one hand, based on the already-in-use scanner data and on the other hand for a proportion of these new hard discounters data partly covered by the product dictionary. Based on previous experiments, the model that has the best performance given the costs is **fastText**. This model, widely embraced within Insee, has several advantages that make it especially suited to point-of-sale data: it has a short training time, is designed to handle noisy texts, including spelling errors, and showcases very interesting performances compared to other state-of-the-art methods. In this study, we measure the proportion of expenditure well classified based on manual labelisation and calculation of confidence intervals. For the automatically labeled part of our new data, the share of expenditure well classified at the National COICOP 6 digits level is above 90%. Regarding the unlabeled part, we developed a specific sampling framework to estimate these performances based on manual verification. Once these first results about classification in the COICOP are validated, some questions remain about using such a tool in production, especially continuous performance evaluation and retraining strategies. The results are more complicated to exploit since we didn't manually classified into the item Coicop 99.9.9.9 unfollowed. If we exclude the data predicted into this category by the model, the share of expenditure well classified at the National COICOP 6 digits level is slightly above 40% and it goes up to 70% if we reduce the scope on the data in which the model is confident in its prediction. This is an ongoing work, improvements and

[1]Harmonised Index of Consumer Prices
[2]Consumer Prices Index
[3]Classification of Individual Consumption by Purpose
[4]European Article Number = bar code

better practice is still needed. [5]

Introduction

- We have new scanner data but current methodology cannot be applied because no structured characteristics are available for these data contrary to "current" scanner data. Hence we cannot classify all the products easily in the COICOP
- Some products are covered, thanks to them we can use machine learning to experiment the possibility of automatic classification.
- In France we work with a more detailed level of COICOP than 5 positions, this is a 6 digit COICOP code, called "postes". This is the target level of classification.
- The objective is to estimate the proportion of expenditure well classified both for the "automatically" classified i.e. the products available in the dictionary and the unlabeled part for which we develop a sampling approach with manual classification.
- The paper is structured as follows : description of data, followed by models and methods used, results and concluding remarks.

Data

Scanner data description

In France, scanner data is used in production to compute our CPI since January 2020. Data was until now provided by all the Super and Hypermarket, hard discounter excluded. We are getting data from retailers, thanks to an article of law originally published in 2017, and modified in 2021, making mandatory for retailers to provide us data for any day and shop, each day. These are used for prices' statistics and turnover indicators. The data requested is the following:

- EAN (European Article Numbering)
- Outlet id
- Date of the sale
- At least two variables among the 3 following: number of article sold, the whole expenditure and the unit price of the article.
- A label, which can be relatively short and rarely exceeds 25 characters (space included)
- The intern nomenclature code given by the retailer

We have now scanner data for two hard discount retailers, from which we extracted the following caracteristics per product:

[5] Many thanks to Julien Peignon and Theo Leroy, who worked on classification topics using scanner data and whose work and help were very useful.

- Label, consisting in a character string without any special structure
- EAN (European Article Number)
- Expenditure for a given period of time. It appears that it is sometimes negative.

We focused our experiment on the retailer for which we had a complete year of data for 2023. It represents 265 672 distinct EAN.

Product dictionary

The law as it is written at the moment implicitly supposes that there is a dictionary that we can use to our purpose of describing and classifying products into a nomenclature. Indeed, as it is written, the only descriptive information is the label. These data are indeed used in our process with a product dictionary (bought from an external firm, Circana previously known as IRI) allowing us to get more information on the data: characteristics of the products and a "family number" or product category identifier. With these data, we are able to classify at a granular level our article (called "variety" in our framework, which is even finer than COICOP on 6 positions – a specific level of France) using classification rules made for each variety to select observations. Moreover, the scope of scanner data used in production is hyper and supermarkets in Metropolitan France, for sales of processed food products, cleaning products and hygiene and beauty products and also, some durable goods. The scanner data expenditure share in the CPI weights is around 10% of the whole basket. We do not use a larger consumption scope because we do not possess information in our referential about these products. Hence, we cannot classify these products, we cannot control for their units, etc. In our internal process, these excluded products are allocated to a specific coicop item "99.9.9.9.9 : unclassified".

The referential is built by our external provider from a field survey and data from manufacturers and distributors that does not entirely include the hardiscount outlets. However some of the products sold in those outlets are found in the referential.

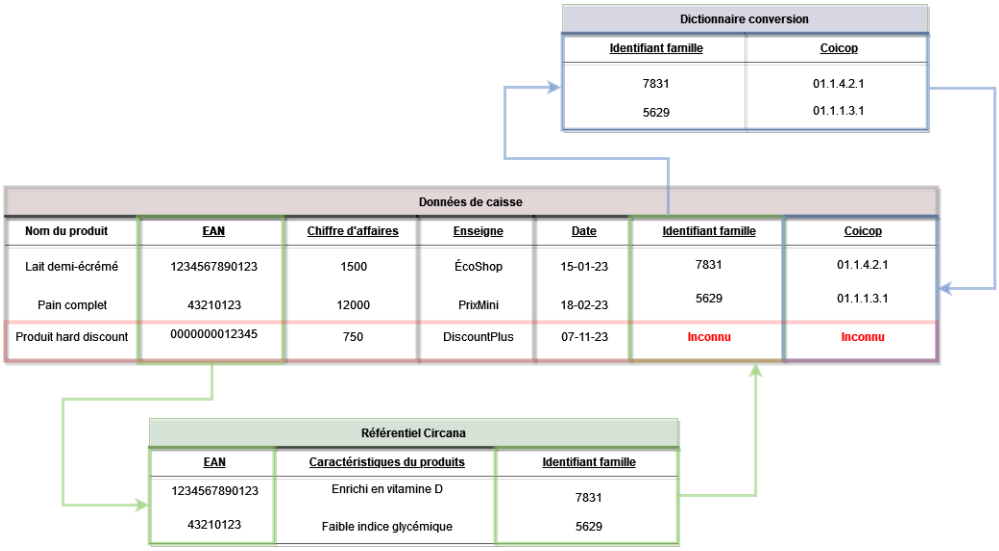

Figure 1: Process to classify scanner data with the dictionary from our external provider

At the moment we also have field collected data that can help us having a better label for the products.

Among the 265 672 EAN, 222 897 (84%) cannot be linked to a product category from the dictionary and therefore automatically linked to COICOP. This important proportion of the products represents 37% of the total expenditure. The reason why the proportions are so different is that the producer of the dictionary probably focuses on the products of hard discounters that are most sold. Some of the EAN can be linked to a product category which isn't classified into COICOP : they are 11 072 and represent 14% of the total expenditure.

If we focus here on the classification into the COICOP based on the dictionary, it is also used to build homogeneous products based on detailed characteristics. If not discussed here, this is a very important feature of our current index calculation methodology.

Data cleaning process

We describe here the main tasks performed during the data cleaning.

Exclusions

We excluded observations with a negative expenditure.

Variable cleaning

- **EAN cleaning** :
  - Some EAN don't respect the constraint of being 8 or 13 digits. If they are of 15 digit, it's often due to the prefix "55" added before, we remove it. Otherwise if another correct EAN with the same label exist, we replace the wrong one by it. Otherwise we complete the EAN by adding 0s before.
  - Some label are the same have multiple EAN. We set all the observations to the same EAN (a correct one if available).
- **Label cleaning** :
  - Convert all characters into ASCII.
  - Removal of stopwords (like articles or pronouns : "le", "la", "et", "ou"), using the library *Spacy*
  - Lemmatization using the library *Spacy* : regroup some words that are alike and replace them by a lemma. There are already word vectors created for a lot of langages [6]. It removes plurals and conjugate at the infinite. Example in French : « chevaux » => cheval. For some words it is odd and uncorrect : « 'poireaux' - leeks» => poireal that does not exist (it is 'poireau').
  - step specific to scanner data : replace information concerning volume or units (weight, size, gender, lot,) by a specific indicator : for instance "500g" will replaced by "#WEIGHT" or in French "#POIDS"

At the end of this, we have a well structured and cleaned data set with:

- Cleaned EAN
- Cleaned label
- Corresponding expenditure amount
- For 16% of the products the corresponding 6 digits coicop code

This is from this dataset that we will train and test automatically our model, and then perform sampling for a manual evaluation of the performance.

Models and Methodology

Classification model

The model used is **fastText**. This model, widely embraced within Insee for other classification tasks, has several advantages that make it especially suited to point-of-sale data : it has a short

[6] https://fasttext.cc/docs/en/crawl-vectors.html

training time, is designed to handle noisy texts, including spelling errors, and showcases very interesting performances compared to other state-of-the-art methods.

fastText is a neural network architecture designed to be resource-efficient. Specifically optimised for minimal use of computing resources, it makes the use of GPUs unnecessary (Joulin et al. 2017). This efficiency is due to pruning, which aims to retain only the essential features of the trained model, and to a quantization of the weight matrices. This model is structured around embeddings, a form of vector representation of words. These have replaced the traditional Bag of Words approach. Unlike the old frequentist methods, which used scattered vectors, embeddings make it possible to represent a word using a dense vector. A notable advantage of embeddings is their ability to synthesise the meaning of a word based on the context in which it appears. As example in scanner data we might encounter terms such as *couette en plume* (feather duvet) and *stylo à plume* (fountain pen). Embeddings recognise that the word feather has a different meaning when associated with *couette*, instead of "stylo*

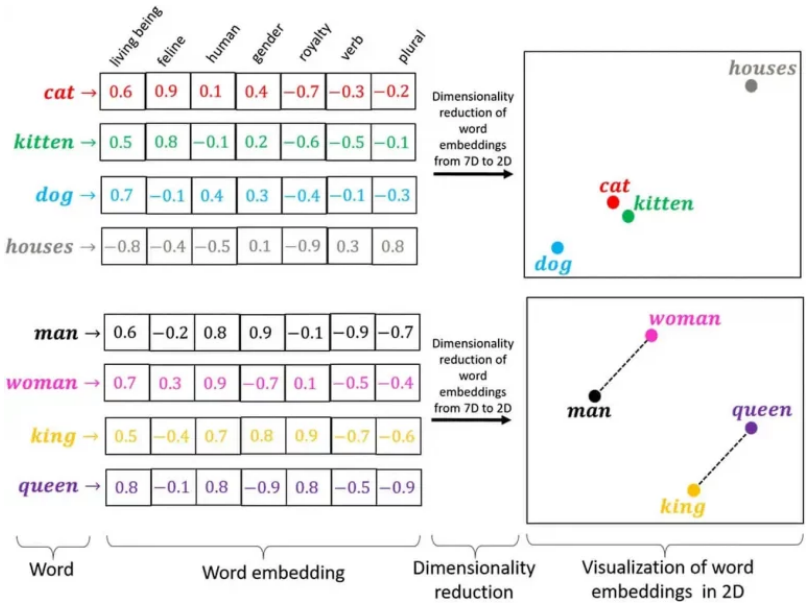

Figure 2: Simplified example of an embedding of words (source : jems-group)

Figure 2 shows a simplified representation of how embedding work. In this example the words are vectorised in a 7-dimensional space. When we reduce this space to visualize the semantic relationships in two dimensions, some notable similarities emerge. For example, the words "man" and "woman" are as close to each other as "king" and "queen" are. This suggests that the model has not only not only grasped the gender relationship between these terms, but also understood that 'king' and 'queen refer to closely related concepts. Thus, words that share similar or close meanings are represented by vectors with similar values in vector space. Embeddings can be calculated using a machine learning model called Word2Vec (see (Mikolov and Dean 2013)). Two variants of this model exist: skip-gram and CBOW (Continuous Bag

of Words). **fastText** offers the option of using either of these two variants of embedding, enriching them significantly by means of n-grams. embedding, significantly enriching them with character n-grams (see (Bojanowski and Mikolov 2002)). Indeed, a notable shortcoming of these models is their inability to handle words that are not present in the training corpus. These approaches consider the word as the fundamental unit of analysis and define a single representative vector. Thus, a word absent from the training corpus remains without any vector representation. This limitation is not present in **fastText**. Rather than treating each as a single entity, **fastText** breaks them up into n-grams of characters. These n-grams are then converted into vectors, which are then aggregated to represent the word as a whole. For example, let's take the word "France" with n set to 3: we get G = {[fr, fra, ran, anc, nce, ce], [france]} as a set of representative 3-grams. This method makes it possible to identify the different declensions of a word in a given language. Although each form of the word may not be present in the corpus, the vectors, once learned, capture and reflect the similarities and common features between these different variations. For instance, **fastText** is able to discern that the words 'cacahuète', 'cacahouète' and 'cacahouette' refer to extremely similar concepts, even if only one of these spellings is present in the model's training data. Thus, **the use of character n-grams proves to be particularly beneficial for developing algorithms that are robust to spelling errors, typos and abbreviations**: a major advantage for in cash register data processing.

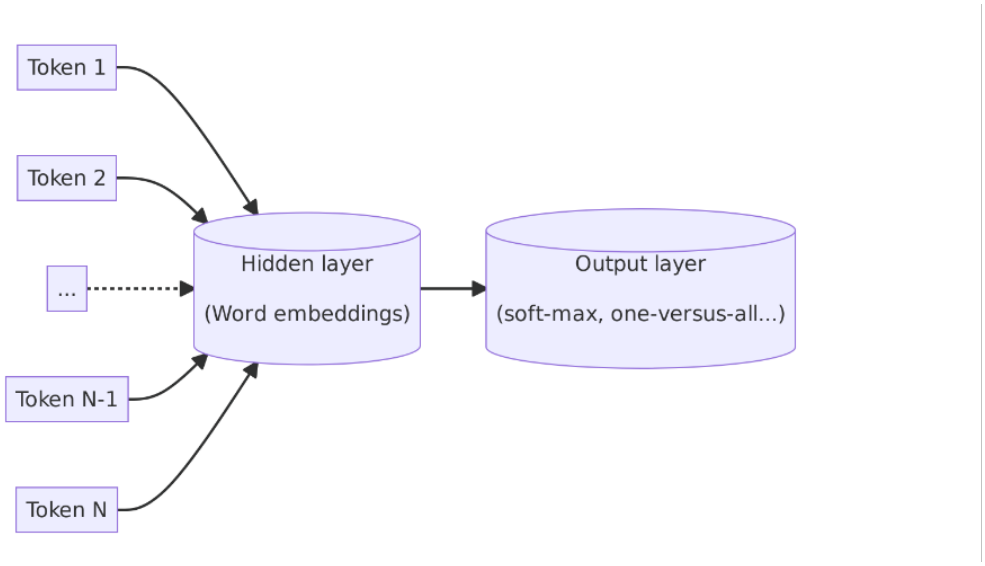

Figure 3: **Fast text** model architecture

The architecture of **fastText**, illustrated in Figure 3, consists of two main layers. The first is a hidden embeddings layer. As we have explained, it converts vectors of tokens - the simplest units of text, such as words or characters - into dense vectors. The second layer is a configurable output layer. Depending on the specific task to be performed, it can be configured in different ways, including, but not limited to, the use of functions such as soft-max or one-versus-all.

Finally, a noticeable advantage of the **fastText** library is its ability to assess the confidence of predictions using probability scores. The latter determines the probability of a text - in this case, the product descriptions - belonging to a set of classes. Confidence scores provide a measure of the certainty of the model when it assigns a COICOP position to a product. In cases where the number of classes is large, **fastText** allows the use of the soft-max hierarchical activation function from (Goodman 2001). This reduces the computational complexity from $O(kh)$ to $O(hlog_2(k))$, where k is the number of classes and h the dimension of the embedding vectors (see (Joulin et al. 2017)). Formally, the probability that a text x belongs to a class k is expressed as :

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{\sum_{i=1}^{K} p(x|C_i)p(C_i)} = \frac{e^{a_k(x)}}{\sum_{i=1}^{K} e^{a_i(x)}}$$

with :

$$a_k(x) = log(p(x|C_k)p(C_k)) = log(p(C_k|x)), \forall k \in \{1, .., K\}$$

Methodology

As explained in the data section, we have some already labeled observations thanks to the product dictionary. These observations is our basis for building our model. As the literature suggests, we define a training and a test sample following a 80%/20% random partition. Based on this, we trained the **fastText** model with the following parameters.

Parameter	Value
Size of word vectors	100
Learning rate	0.1
Number of training epochs to train for	100
Number of word n-grams to consider during training	3
length of char ngram	3 to 6
Min number of word occurences to be in the dictionary	3
Number of buckets	$1,000,000$
Type of loss	soft-max

After the model is trained, we can use it to predict the COICOP item of our labels. In the output, we get a list of prediction and their associated probability. It has to be noted that this is not strictly a probability since there is for each possible code a corresponding "probability" whose sum is not systematically equal to 1. Yet, the difference of the two highest "probability" can be used to build a confidence measure in the prediction. Our main goal is to check the global and local (from a Coicop perspective) perforamnce of the model. Also, we are interested to see if when the model is confident, the classification is good and vice-versa. The lowest level at which we broadcast our indexes is COICOP 6 digits. We chose to train the model and use

the model to predict this level.

We allowed the model to predict into the custom coicop item "99.9.9.9.9" since it was part of the train set.

Performance on already classified, the "test sample" of the labelled dataset

The easiest way to test the ML model is to test it on the "test sample" we set aside before training the model. We will look at the share of expenditure well classified, the share of observation well classified and also make a focus on the data on which the model has a good confidence. Our working level is COICOP 6 Digits.

Sampling for manual classification and confidence interval calculation

In practice, we would use the model on the data we cannot currently classify into COIOCOP. Therefore, it is of importance to test the performance of the model on these data. We are interested to measure the expenditure share of the scanner data that is correctly classified by the model:

$$R = \frac{\sum_{k \in U} CA_k \times z_k}{\sum_{k \in U} CA_k}$$

where:

- k is an article.
- U represents the sampling universe of the articles (represented by their EAN) sold during the year 2023.
- CA_k is the cumulative expenditure of the article k in our scanner data in 2023.
- $z_k \in \{0, 1\}$ whether or not the EAN is classified into the right COICOP item (level to be defined).

Even if it is not our main goal, the share of observations correctly classified by the model could be another helpful indicator.

Since there are more than 200 000 EAN sold during the year 2023 that are not found in the product dictionary, it is unfeasible to manually label each product. Then, our approach was to define a sample to be labeled manually and then test our model accuracy and be able to calculate a confidence interval.

The sample size was defined according to the workforce available for this manual annotation: a sample size of 3 000 labels to annotate. Based on this, we sampled the products to be labelled according to a stratified sampling.

1000 distinct EAN represent more than 50% of the expenditure (Figure 4).

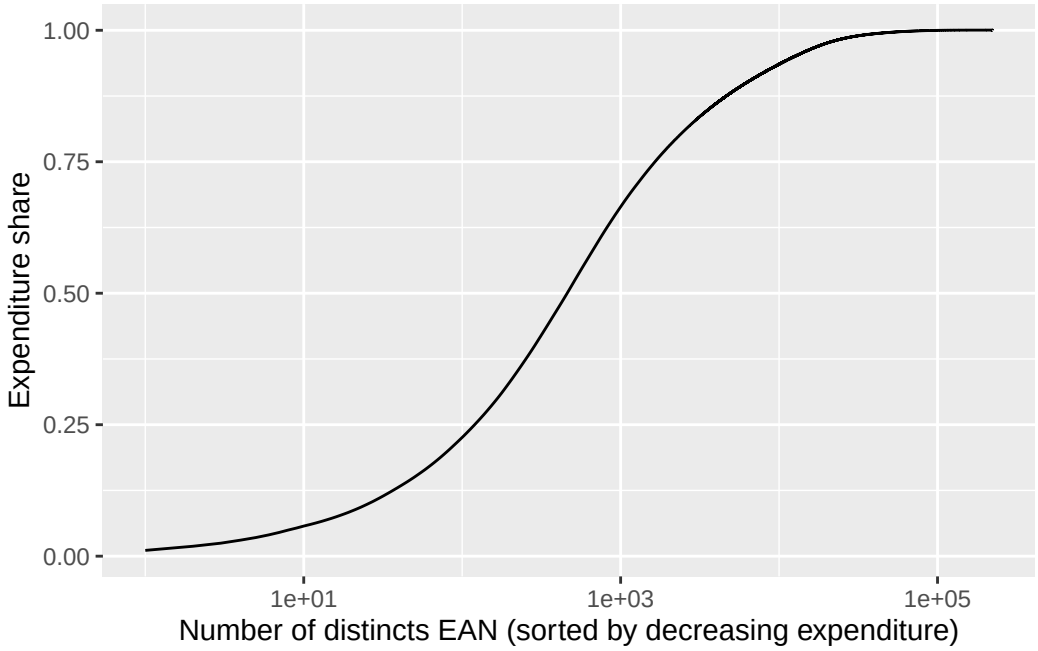

Figure 4: Expenditure share according to the number of obervation.

Our strategy was to stratify according these two variables in order to minimize variance in each stratum:

- the amount of expenditure the product represents
- an indicator of the **confidence of the model in its prediction (the difference of the two best prediction probabilities for each label)**.

The expenditure share of likely to be misclasified products is quite high (more than 30%) according to Figure 5 which is a bad sign.

We then defined the size of the sample in each stratum according to its expenditure share The sample of 3 000 products to label manually was distributed as follows (Table 2)

Table 2: Sample Distribution

expenditure	Confidence of the model prediction	Sample size	Number of EAN in the stratum	Sampling ratio
[0,5e+04)	[0,0.1)	177	71429	0.25 %
[0,5e+04)	[0.1,0.9)	116	75665	0.15 %
[0,5e+04)	[0.9,1]	104	61097	0.17 %
[5e+04,2e+06)	[0,0.1)	844	4494	18.78 %
[5e+04,2e+06)	[0.1,0.9)	651	2489	26.16 %

Table 2: Sample Distribution

expenditure	Confidence of the model prediction	Sample size	Number of EAN in the stratum	Sampling ratio
[5e+04,2e+06)	[0.9,1]	394	1556	25.32 %
[2e+06,7.32e+07]	[0,0.1)	209	209	100 %
[2e+06,7.32e+07]	[0.1,0.9)	266	266	100 %
[2e+06,7.32e+07]	[0.9,1]	240	240	100 %

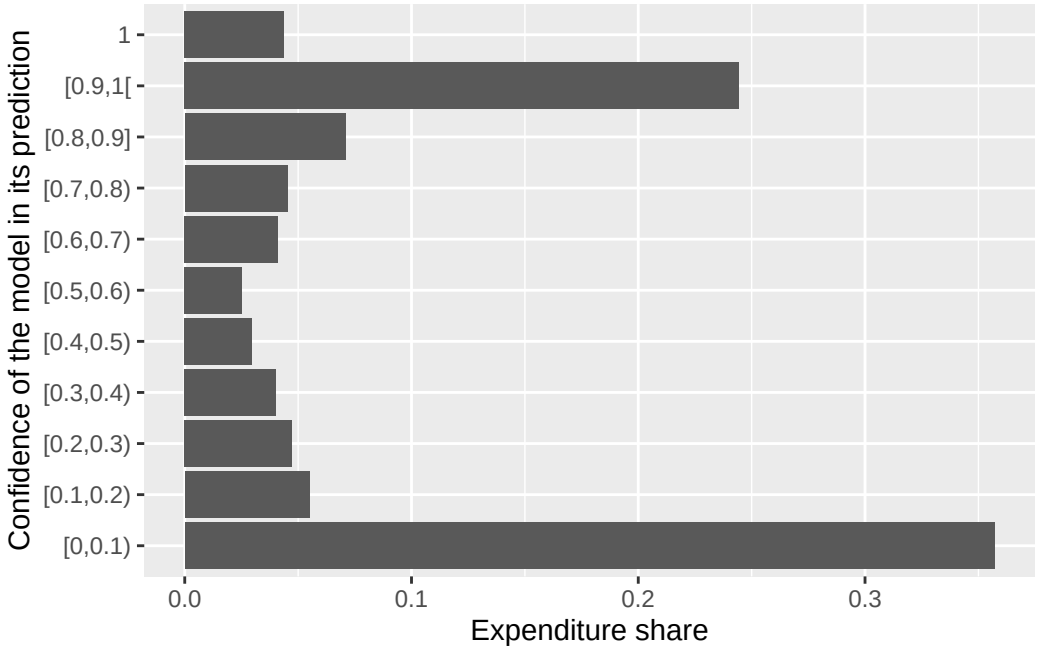

Figure 5: Distribution of expenditure according the confidence of the model in its prediction

With given samples for each stratum S_h, we defined our estimator of $\widehat{R}$ for the total sample S as

$$\widehat{R} = \sum_h W_h \widehat{R}_h$$

where :

- $W_h = \frac{\sum_{k \in U_h} CA_k}{\sum_{k \in U} CA_k}$ is the share of expenditure of the stratum h
- $\widehat{R_h} = \frac{\sum_{k \in S_h} CA_k z_k}{\sum_{k \in S_h} CA_k}$ is estimator of the share of expenditure of the stratum h well predicted

An approximatively unbiased estimator of $\text{Var}(\widehat{R})$ is:

$$\hat{V}(\hat{R}) = \sum_{h=1}^{H} W_h^2 \widehat{V}(\widehat{R}_h) = \sum_{h=1}^{H} W_h^2 N_h^2 \frac{1 - f_h}{n_h} s_{u,h}^2$$

with $f_h = n_h / N_h$, the sampling rate for stratum h, and $\forall h \in \{1, \dots, H\}, \forall i \in S_h$

- $\widehat{u_i} = \frac{CA_i z_i - R_h \hat{C}A_i}{\widehat{t_{h,D}}}$
- $\widehat{\mu_{u,h}} = \frac{1}{n_h} \sum_{k \in S_h} \widehat{u_k}$

- $s^2_{u,h} = \frac{1}{n_h - 1} \sum_{k \in S_h} \widehat{u_k} - \widehat{\mu_{u,h}}$

This is approximate because of linearisation of the $\widehat{R}_h$ around R_h :

$$\widehat{R}_h = \frac{\sum_{k \in S_h} CA_k z_k}{\sum_{k \in S_h} CA_k} \approx R_h + \frac{\sum_{k \in S_h} CA_k z_k - CA_k R_h}{\sum_{k \in U_h} CA_k}$$

since R_h is deterministic, $V(\widehat{R}_h) \approx V(\frac{\sum_{k \in S_h} CA_k z_k - CA_k R_h}{\sum_{k \in U_h} CA_k})$ which is estimated by $\widehat{V}(\widehat{R}_h) = N_h^2 \frac{1 - f_h}{n_h} s^2_{u,h}$

The corresponding (estimated) confidence interval à the 95% rate is the following:

$$CI_{0.95} = \widehat{R} \pm 1.96 \times \sqrt{\widehat{V}(\widehat{R})}$$

Manual classification in practice

After the sample was drawn, we classified the observation using a tool called **LabelStudio**, see Figure 6 for a screenshot of the tool.

- 10 persons participated into the process, classifying between 200 and 400 observation each. Less than 3 days were necessary to classify the whole sample.
- Each observation was only classified by one person, no double check was set in place.

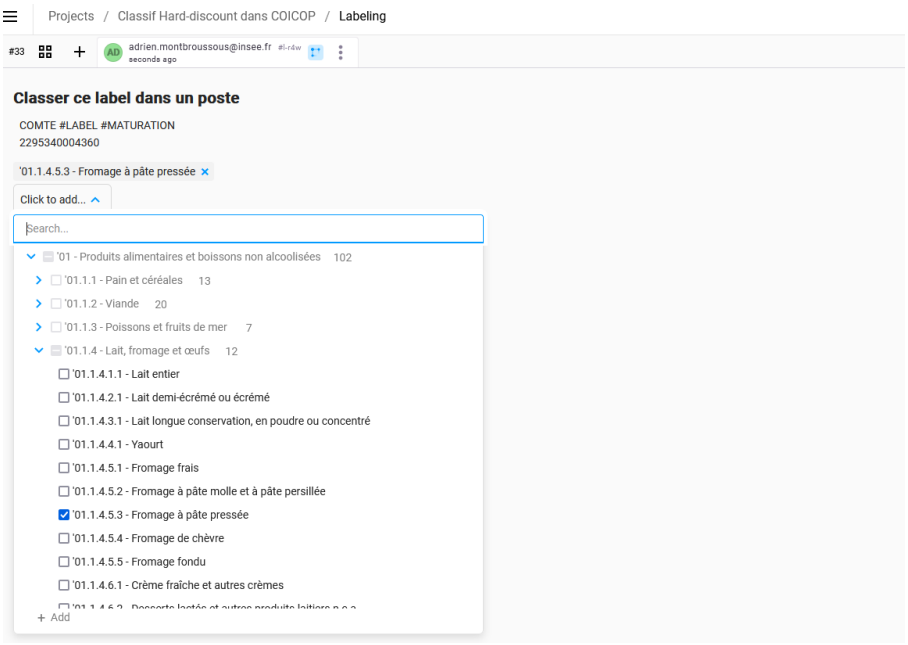

Figure 6: Label Studio screenshot with the use of a taxonomy for COICOP

Results

Performance on already labelled data (test sample)

The model has been trained at the 6 digit level and predictions are done at this level. We won't analyzed performance at more aggregated levels for already labeled data since the results are already quite good.

COICOP 6 digit level ("Postes")

- With the "unfollowed" (the products falling into the `99.9.9.9.9` COICOP item)

Confidence of the model prediction	Share of expenditure	Share of expenditure well classified	Share of observation well classified
[0,0.1)	0.65 %	36.6 %	47.5 %
[0.1,0.9)	9.38 %	87.52 %	81.89 %
[0.9,1]	89.96 %	98.84 %	98.92 %
TOTAL	100%	97.37	96.58 %

- Without the "unfollowed" (the products falling into the `99.9.9.9.9` COICOP item)

Confidence of the model prediction	Share of expenditure	Share of expenditure well classified	Share of observation well classified
[0,0.1)	0.68 %	40.25 %	48.04 %
[0.1,0.9)	10.81 %	87.17 %	83.41 %
[0.9,1]	88.51 %	99.12 %	98.99 %
TOTAL	100%	97.43	96.42 %

Performance on "unlabeled" data

The model has been trained at the 6 digit level and predictions are done at this level. We will also analyzed performance at more aggregated levels as well (without training again our model).

COICOP 6 digit level ("Postes")

The more interesting metric for checking the performance of the model is the COICOP 6 digit level, which is the lowest level at which we are publishing our indexes.

- With the "unfollowed" (the products falling into the `99.9.9.9.9` COICOP item):

Confidence of the model prediction	Share of expenditure	Share of expenditure well classified
[0,0.1)	30.28 %	10.9 ± 1.75 %
[0.1,0.9)	37.26 %	31.83 ± 1.91 %
[0.9,1]	32.46 %	34.58 ± 1.69 %

Share of expenditure well classified
25.15 ± 1.04 %

- Excluding the "unfollowed" (the products falling into the `99.9.9.9.9` COICOP item):

Table 7: Expenditure share well classified at the COICOP 6 digit level according to the confidence of the model in its prediction

Confidence of the model prediction	Share of expenditure	Share of expenditure well classified
[0,0.1)	34.36 %	15.6 ± 2.35 %
[0.1,0.9)	38.74 %	50.05 ± 2.79 %
[0.9,1]	26.9 %	72.36 ± 2.49 %

Reading Note : The data in which the model has a very low confidence in its prediction (<0.1) represents 34.36 % of the expenditure. Among this data, only 15.6 ± 2.35 % is classified into the right COICOP 6 digit item.

The global well-classified proportion:

Share of expenditure well classified
41.44 ± 1.51 %

COICOP 5 digit level (Sub-Classes)

- With the "unfollowed" (the products falling into the `99.9.9.9.9` COICOP item):

Confidence of the model prediction	Share of expenditure	Share of expenditure well classified
[0,0.1)	30.28 %	21.22 ± 2.37 %
[0.1,0.9)	37.26 %	46.31 ± 2.78 %
[0.9,1]	32.46 %	40.7 ± 1.7 %

Share of expenditure well classified
35.73 ± 1.39 %

- Excluding the "unfollowed" (the products falling into the `99.9.9.9.9` coicop item):

Confidence of the model prediction	Share of expenditure	Share of expenditure well classified
[0,0.1)	30.28 %	30.69 ± 2.41 %
[0.1,0.9)	37.26 %	73.91 ± 3.18 %
[0.9,1]	32.46 %	85.11 ± 3.99 %

The global well-classified proportion:

Share of expenditure well classified
61.7 ± 1.83 %

COICOP 4 digit level (Classes)

- With the "unfollowed" (the products falling into the `99.9.9.9.9` COICOP item):

Confidence of the model prediction	Share of expenditure	Share of expenditure well classified
[0,0.1)	30.28 %	39.17 ± 2.45 %
[0.1,0.9)	37.26 %	57.79 ± 2.71 %
[0.9,1]	32.46 %	47.04 ± 1.72 %

Share of expenditure well classified
48.04 ± 1.39 %

- Excluding the "unfollowed" (the products falling into the `99.9.9.9.9` COICOP item):

Confidence of the model prediction	Share of expenditure	Share of expenditure well classified
[0,0.1)	30.28 %	56.48 ± 2.48 %
[0.1,0.9)	37.26 %	91.6 ± 3.1 %
[0.9,1]	32.46 %	97.75 ± 4.18 %

The global well-classified proportion:

Share of expenditure well classified
80.83 ± 1.86 %

COICOP 2 digit level

The classification at the highest level according to COICOP divisions might be useful to understand the mistakes made by the model.

- with the unfollowed:

Confidence of the model prediction	Share of expenditure	Share of expenditure well classified
[0,0.1)	30.28 %	43.15 ± 2.54 %
[0.1,0.9)	37.26 %	59.28 ± 2.68 %
[0.9,1]	32.46 %	47.31 ± 1.72 %

The global well-classified proportion:

Share of expenditure well classified
50.07 ± 1.4 %

- without the unfollowed:

Confidence of the model prediction	Share of expenditure	Share of expenditure well classified
[0,0.1)	34.36 %	62.64 ± 3.39 %
[0.1,0.9)	38.74 %	93.56 ± 1.05 %
[0.9,1]	26.9 %	98.73 ± 0.79 %

The global well-classified proportion:

Share of expenditure well classified
82.39 ± 1.42 %

Analysis of the predictions at the COICOP 2 digit level

The two following diagrams show at the 2 digit level where the products were classified. On the left is the COICOP 2 digits level from manual classification and on the right is the COICOP 2 digits level predicted by the model.

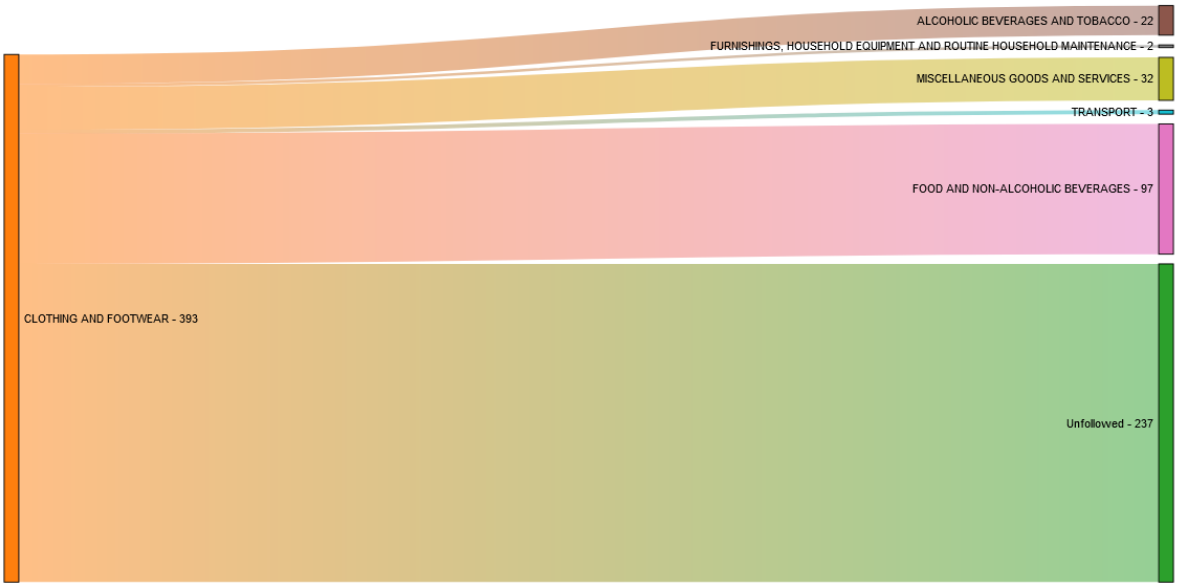

Figure 7: Classification of 'Clothing and footwear'

We currently don't follow with our scanner data clothing and footwear products. The model was only trained with 'unfollowed' classification for these product whereas with manual classification we where able to identify them. We can see that the model correctly classified half of them into the category "unfollowed" but an important number of them is classified into food.

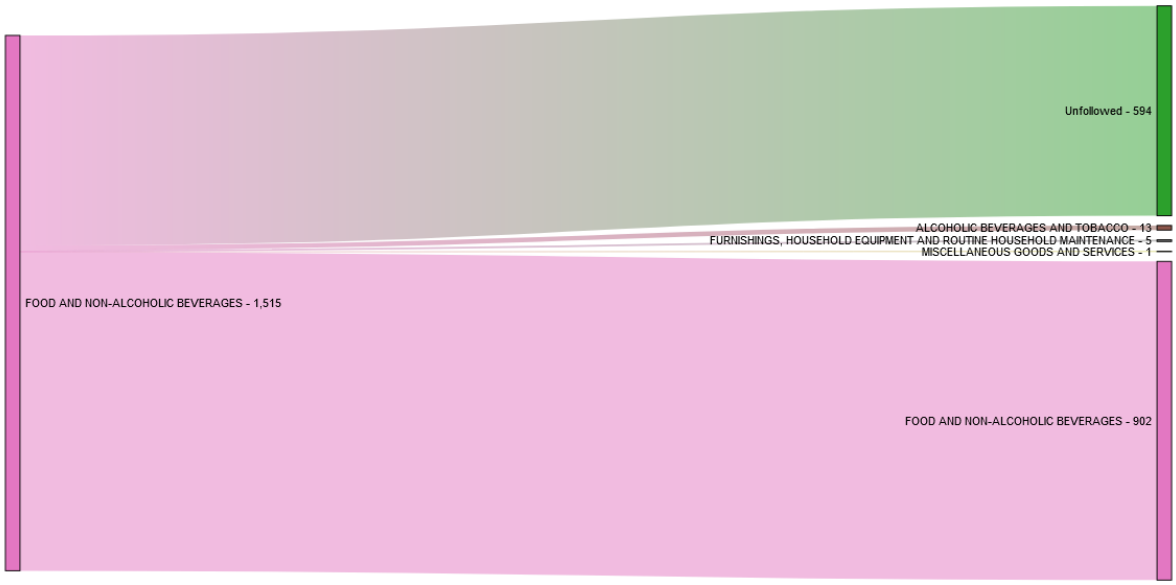

Figure 8: Classification of 'Food and non-alcoholic beverages'

Food items are mainly classified into the right division or into '99 - unfollowed' by the model. However, the model wrongly classify other product into the division " 01 - FOOD AND NON-ALCOHOLIC BEVERAGES "

The following results (Figure 9) are to take into account with even more precautions than the previous, since the number of observation for each COICOP 2 digit level might be small. Moreover, neither the information whether or not the model is confident on the prediction nor the expenditure of each article aren't taken into account in the results shown. However, it can be useful to understand where the model has some troubles.

We aggregated for each level by COICOP 2 digits the proportion of products well classified:

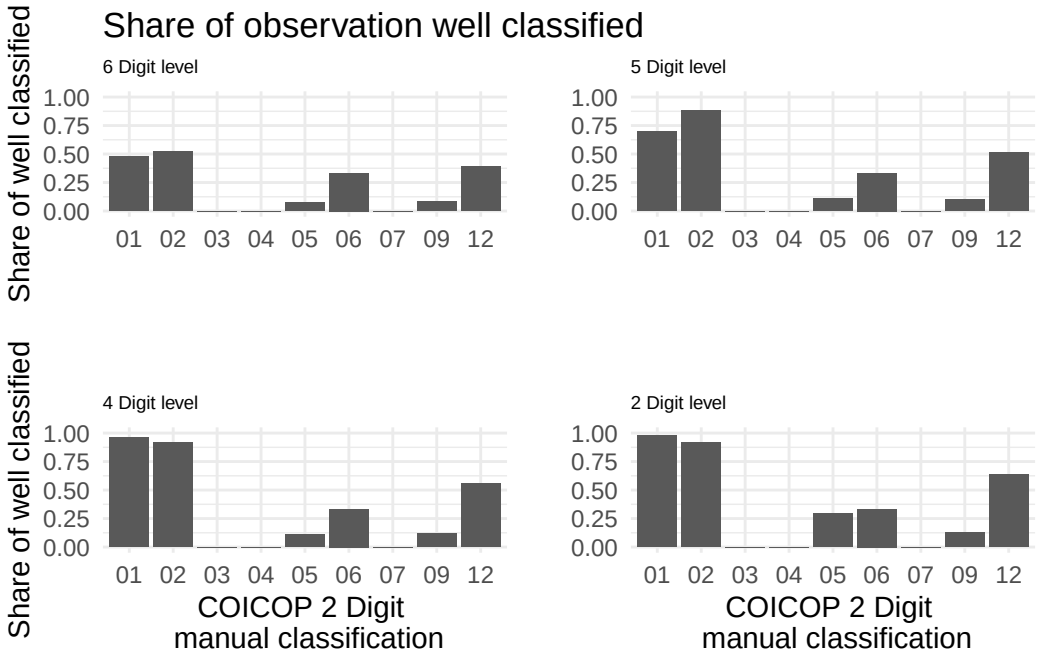

Figure 9: Share of observation well classified, according to several COICOP level and for each COICOP 2 digit item

With :

COICOP 2 Digit	Label
01	FOOD AND NON-ALCOHOLIC BEVERAGES
02	ALCOHOLIC BEVERAGES AND TOBACCO
03	CLOTHING AND FOOTWEAR
04	HOUSING, WATER, ELECTRICITY, GAS AND OTHER FUELS
05	FURNISHINGS, HOUSEHOLD EQUIPMENT AND ROUTINE HOUSEHOLD MAINTENANCE
06	HEALTH
07	TRANSPORT
08	COMMUNICATION
09	RECREATION AND CULTURE
10	EDUCATION
11	RESTAURANTS AND HOTELS
12	MISCELLANEOUS GOODS AND SERVICES

Conclusions

About the results

- Global results at our targeted level are not satisfying at this stage: only a bit more than 40% of expenditure is well-classified.
- Analysis of the level of the COICOP at which performance decreases a lot ¨
- The gaps between the performance of the classification at 4, 5 and 6 digit level are quite important, with respectively around 40%, 60% and 80 % of expenditure well classified (excluding the one classified into the category 'unfollowed').
- It is hard to think of using this approach in a production context. The impact on indexes of this rate of misclassification is yet to be studied but most likely to be important.

About our prediction strategy

- Even though there is a logical explanation to the existence of a "99" division, a finer definition of the classifying rules based on the product dictionary could lead to better prediction
- Removing impossible to classify products (like products labeled "non food") at the beginning could help the model.
- Certain label cleaning steps are counterproductive : the gender written in some clothing products is replaced by a "#gender" tag which does not allow us to classify in the right 6 digit COICOP code.
- In our training data we do not have data for each division (like clothing for instance) and therefore, the model can't classify into these.
- Manual classification before training the model could be useful?

About manual labelisation strategy

- Developing knowledge of the nomenclature is necessary to be efficient and precise in the manual verification.
- Double annotations could be useful to identify easy to classy products and more difficult ones.
- Issues on specifics articles have to be analyzed, for example:
  - fresh vs frozen fish
  - gender clothing
  - wine quality

References

Alexandrescu, Andrei, and Katrin Kirchhoff. 2006. "Factored Neural Language Models." *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume : Short Papers, Association for Computational Linguistics New York City.*

Bojanowski, Edouard Grave, Piotr, and Tomas Mikolov. 2002. "Enriching Word Vectors with Subword Information." *Transactions of the Association for Computational Linguistics, June 2017*, 135--146.

Conneau, Holger Schwenk, Alexis, and Yann Lecun. 2017. "Very Deep Convolutional Networks for Text Classificatio."

Dongmo-Jiongo, Valéry. 2021. "Innovative Uses of Web Scraped Data in the Canadian Clothing and Footwear Consumer Price Index." *High Level Group on the Modernization of Official Statistics: Machine Learning 2021 Monthly Meeting Group.*

Goodman, Joshua. 2001. "Classes for Fast Maximum Entropy Training."

Greenhough, Hazel Martindale, Liam, and Helen Sands. 2022. "Modernising the Measurement of Clothing Price Indices Using Web-Scraped Data: Classification and Product Grouping." *17th Meeting of the Ottawa Group. Rome, Italy.*

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. "Bag of Tricks for Efficient Text Classification." In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 427–31. Association for Computational Linguistics.

Mikolov, Ilya Sutskever, Tomas, and Jeffrey Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." *Transactions of the Association for Computational Linguistics, June 2017.*

Sebastiani, Fabrizio. 2002. "Machine Learning in Automated Text Categorization." *ACM Computing Surveys*, 521--528.

Sun, Aixin, and Ee-Peng Lim. 2001. "Hierarchical Text Classification and Evaluation." *Proceedings 2001 IEEE International Conference on Data Mining*, 521--528.

Technical report, UK Statistical Authority. 2019. "Guidelines for Selecting Metrics to Evaluate Classification in Price Statistics Production." Advisory Panel on Consumer Prices – Technical. 2019. https://uksa.statisticsauthority.gov.uk/wp-content/uploads/2019/08/APCP-T1910-Classification-metrics-guidelines.pdf.

UNECE. 2021. "Machine Learning for Official Statistics. Geneva: United Nations Economic Commission for Europe." Unece Meeting. 2021. https://unece.org/statistics/publications/machine-learning-official-statistics.

William Spackman, Christian Ritter, Greg DeVilliers. 2023. "Identifying and Mitigating Misclassification: A Case Study of the Machine Learning Lifecycle in Price Indices with Web-Scraped Clothing Data." *Unece Meeting.*

Zhang, Junbo Zhao, Xiang, and Yann LeCun. 2016. "Character-Level Convolutional Networks for Text Classificationl Networks for Text Classification."