Apply ML techniques to classification and aggregation of web scraped price data

Vladimir G. Miranda,[*] Lincoln T. da Silva[†]

Brazilian Institute of Geography and Statistics
IBGE
Price Indices Coordination
Directorate of Surveys

[*] vladimir.miranda@ibge.gov.br
[†] lincoln.silva@ibge.gov.br

Contents

1 Background

The Brazilian Institute of Geography and Statistics (IBGE) has recently started to explore the use of new data sources to produce price statistics and improve its work routines [da Silva et al., 2019, Oliveira et al., 2021, Miranda et al., 2021]. Studies on different uses of web data for different sectors of the CPI basket are in course [da Silva et al., 2019, Oliveira et al., 2021] and for a few sectors of the consumer price index (CPI) basket, manual collection has been replaced by an automatic one [da Silva et al., 2019].

The initial focus was devoted to a targeted web scraping approach [Eurostat, 2020] where a predefined list of products of interest can be selected from the data extracted by the scrapers. Though this approach is more parsimonious in relation to the CPI structure it does not explore all the power of big data sources and is hard to scale without the use of automatic tools.

To expand the use of alternative data sources at larger scales, a bulk [Eurostat, 2020] approach is usually employed where a huge volume of products is extracted from these sources and the most of them incorporated in the CPI calculations. Since in this case products of interest are not defined beforehand, in general one is not able to know to which category of the CPI the products of the data set belong. Manual association in this case can be very resource-intensive due the large amount of data. This problem becomes even more intricate due market dynamics which lead to products constantly entering and leaving the market. Besides, the challenge of correctly classifying the new products and keep track of comparable products along time are also a point of major concern that require special methodological treatment [Chessa, 2019].

For the incorporation of a massive number of products via alternative data sources in the CPI routines, automatic tools are necessary to treat and classify the data of interest according to the CPI classification structure. Though different sectors of the CPI basket might require different classification methods [ONS, 2021], machine learning (ML) has been shown to provide powerful tools in this field and the number of CPI compilers considering the use of these techniques is increasing [Myaklatun, 2019, Harms and Spinder, 2019, ONS, 2021, Kruczek-Szepel and Piatkowska, 2020].

Since we had no previous experience with machine learning techniques, the aim of this project was twofold: try to develop some expertise in this field via work on an applied problem and start the construction of tools that are necessary in the process of expansion for the use of alternative data sources at IBGE. We note that currently we do not have access to scanner data sources yet and we focus our efforts on data obtained from the web. However, other important point of this initiative is that once we have access to scanner data, classification issues will also be present [Myaklatun, 2019] and we will already have a machinery that can be used in this case too.

Figure 1 displays the classification system structure adopted at IBGE. It is a four level structure where the lowest level is denoted as *Subitem*. As one move from top to bottom in the structure the level of homogeneity of the elements in each stage increases.

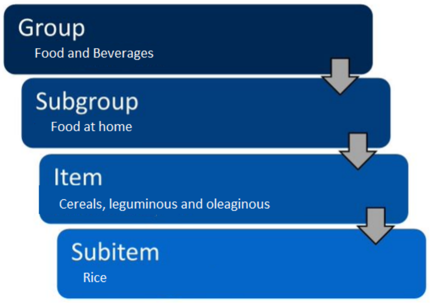


Figure 1: Classification system adopted for the CPIs produced at IBGE.

Products need to be allocated below the subitem to each they belong. In the manual collection process this identification results as a by product of the process of sample selection and maintenance for the subitems since the field collectors will look for a product that fits a given subitem.

In an automatic collection in general occurs the opposite, we have the products beforehand and need to find out to what subitem they belong. This is the problem that we addressed here via use of ML tools.

As we mentioned we had no previous experience with machine learning and during our journey in this year we started the project by first performing a literature review on the experiences of other countries regarding the use of automatic tools for product classification under the scope of CPIs. Among some interesting references we can cite Refs. Martindale et al. [2019], Myaklatun [2019], Harms and Spinder [2019], ONS [2021], Kruczek-Szepel and Piatkowska [2020]. This search was useful so that we could have an overall idea of the tools (both software and models) most used in this area and a clearer idea of the complexity of the problem.

We benefited a lot from the work conducted by colleagues from Stats Poland Kruczek-Szepel and Piatkowska [2020] which worked in a similar problem in the 2020 round of the UNECE ML group. Besides providing the report, the authors also shared the codes they developed in the project which was very beneficial for us. In our study we reproduced some of the approaches adopted by these authors.

Though some works [Martindale et al., 2019] make use of more sophisticated approaches, the models used by the authors in Ref. Kruczek-Szepel and Piatkowska [2020] are also adopted for studies by other countries [ONS, 2021] and in some cases are even being used in current production [Myaklatun, 2019].

In the following we describe the main aspects of the work conducted.

2 Data

2.1 Input data

During the course of this year, we have reported our advancements in the many phases of this project and among them we mentioned that we have considered different data sets in our studies. As we did not have large data sets of web scraped data encompassing a great number products covering a large number of categories of the CPI basket, we had to start our studies with data sets that we had at hand.

We first started our studies with web scraped data that only considered a few categories of electronics and household appliances. This was composed by data that we have been collecting by scrapers developed by our own for the purpose of other studies [da Silva et al., 2019]. Unfortunately, as the number of categories was reduced, the first model tested (logistic regression) had no problem in classifying the data set with 100% precision.

In the second quarter and part of the third quarter of the year we evaluated the classification of product descriptions that are part of the current frame of products currently considered for the CPI collection. This data set was much more realistic than the first one we considered, hence allowed us to evaluate the use of more complex preprocessing techniques and other machine learning models.

From the third quarter on we started to get access to larger volume of web scraped data from different product categories and web sites. These results are obtained from scrapers developed under the scope of a collaboration between IBGE and researchers from the Federal University of Minas Gerais (UFMG). This is an ambitious project initiated during the pandemic that intended to scale up the use of web scraping to a larger number of categories of the CPI basket via collection in different stores. A set of modules were developed in python to reach this goal.

The results presented in the main text below take into account this latter source. They cover the whole set of preprocessing and ML models we evaluated during the year. Nevertheless, some of the results presented in the other quarters can be found in the appendices of this report.

The whole data set considered consists of web scraped products from over 300 online stores. However, as we need to perform some manual label of the data sets in order to build our training sets, we restricted our analysis to a subset containing only grocery products. This led to a set with 8 974 unique product descriptions. An example of how such descriptions look like and their relation to a given subitem is presented in Table 1.

The data covers 135 subitems from the CPI basket associated to groceries. Figure 2 shows how the products descriptions obtained are distributed across these different subitems. As one can note the distribution is quite unbalanced with the number of observations for some categories differing by orders of magnitude as shown below. Hence, this can have an important impact on the performance of the algorithms to correctly classify some of the classes.

Table 1: Examples of how product descriptions and subitems look like in the data set.

Product description	**Subitem**
arroz agulhinha tipo 1 camil pacote 1kg	Arroz
arroz agulhinha tipo 1 camil pacote 5kg	Arroz
feijão carioca tipo 1 pantera premium pacote 1kg	Feijão-carioca (rajado)
feijão carioca tipo 1 qualitá pacote 1kg	Feijão-carioca (rajado)
macarrão dallas espaguete speciallitá pacote 500g	Macarrão
macarrão dallas espiral com ovos pacote 500g	Macarrão
farinha de trigo tradicional qualitá pacote 1kg	Farinha de trigo
farinha de trigo tradicional renata tipo 1 pacote 1kg	Farinha de trigo
açúcar refinado caravelas pacote 1kg	Açúcar refinado
açúcar refinado especial da barra 1kg	Açúcar refinado

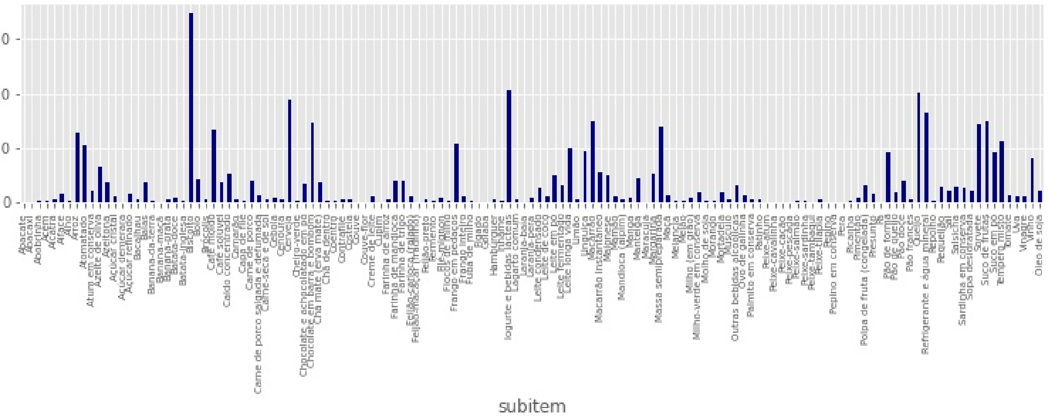


Figure 2: Distribution of product descriptions according different categories for grocery subitems.

2.2 Data preparation

In the following we describe all the data preparation steps considered in the study.

2.2.1 Normalization

Normalization is the process of putting a document into a standard way. Among different procedures that can be implemented to reach this we here consider:

- **Word tokenization**: which consists in splitting the text into words.
- **Case folding**: words were converted into lower case. The classification for our purposes is not case sensitive (e.g., Rice → rice).
- **Punctuation removal**.

2.2.2 Natural Language Processing (NLP)

We evaluated the use of different preprocessing treatments to deal with the natural language aspect of the product descriptions. The treatments that were tested may include:

- **Lemmatization** consists in transforming the word to its root in spite of its suffix, prefix, etc. (e.g., he is eating rice → he be eat rice)

- **Stemming** cuts the words except for its root. It is a simpler method than lemmatization. (e.g., this was complete → thi wa complet).
- **Stop words removal**: Stop words are words that do not add much information to the text. They are usually represented by articles, prepositions, auxiliary verbs.

For both lemmatization and stemming, we used python's spaCy library [Honnibal and Montani, 2017]. We selected the stop words for our case via use of the nltk python's library [Loper and Bird, 2002] for Portuguese language.

2.2.3 Feature Engineering

Vectorization

Other basic step while dealing with natural language processing is to convert the strings into a numerical representation which will be used as an input for the ML models. This process is known as vectorization. We checked two approaches of vectorization, namely, countvectorizer and tf-idf from python's sklearn library [Pedregosa et al., 2011].

Countvectorizer builds a $d \times w$ matrix with the raw counts of the token occurrences in the document. Here, w stands for the different words or tokens found for all d descriptions. Table 2 shows a counvectorizer example.

Table 2: Example of how CountVectorizer transforms descriptions into numerical vectors. The matrix entries for a given row counts the number of times the token appears in the description associated to it.

text	beans	brandA	brandB	rice	size1	size2
rice brandA size1	0	1	0	1	1	0
rice brandB size1	0	0	1	1	1	0
beans brandB size2	1	0	1	0	0	1

It happens that texts may contain high frequency words with meaningful information. Countvectorizer does not consider this into account and attributes the same weight to a word irrespective of its relative importance.

There are different approaches that try to remedy this by ascertaining weights according to the relevance of the words contained in the text. One of such techniques is the so-called term frequency-inverse document frequency (Tf-idf).

The main idea behind the Tf-idf is to re-weight the raw counts by the product of two terms: Term Frequency and Inverse Document Frequency. The former measures how frequently a term occurs in a a product description while the latter calculates the ratio between the total number of product descriptions considered and the number of product descriptions in which the term appears. This means that the terms that appear in multiple documents will have a greater relevance.

As the countvectorizer, the tf-idf also returns a $d \times w$ matrix as illustrated in Table 3.

Table 3: Example of vectorization after implementing the tf-idf procedure.

text	beans	brandA	brandB	rice	size1	size2
rice brandA size1	0	0,68	0	0,52	0,52	0
rice brandB size1	0	0	0,58	0,58	0,58	0
beans brandB size2	0,62	0	0,47	0	0	0,62

n-gram

Sometimes words apart do not make as much sense as a sentence of n words combined. Furthermore, they can have different meanings when separated. For instance, "olive oil" compared to just "olive" and "oil".

A technique called **n**-gram is usually applied in natural language problems to deal with this. The reasoning behind a word **n**-gram is to generate tokens composed of n words that appear in a sequence in an expression. An illustration of how **1**-gram and **2**-gram representation look like for the text 'rice brandA size1' are exemplified below.

- **1**-gram gives three tokens: 'rice', 'brandA', 'size1'.
- **2**-gram gives two tokens: 'rice brandA', 'brandA size1'.

This work evaluates the use of **n**-gram up to 2-gram (also denoted as bigram), i. e., we evaluated 1-gram alone and the combination of 1-gram and 2-gram in our preprocessing steps.

2.3 Feature Selection

The matrix generated by feature engineering after data preparation gives the features for the models.

2.4 Output Data

The output are the categories associated to each description for a given model and the probabilities of the assignments made.

3 Machine learning solution

As a start we split the data set into training and test sets. As usual, the reason for this is that the model will by fitted in the training set then its performance is evaluated via the test set.

The models were estimated via GridSearchCV that exhaustive search the best score (accuracy) over all hyperparameter combinations with 5-fold for the cross-validation. The advantage of the cross-validation approach is that no validation set is required when splitting the data set. The use of this technique avoids reducing the number of samples that can be used in training and test sets. Furthermore, the optimal hyperparameters do not depend on a particular train and validation sets.

The split proportion used is 60% and 40% for the training and test sets, respectively. Under this scenario, the number of rows for the training set is 5 384 and for the test set is 3 590.

3.1 Models tried

We evaluated the performance of several ML models for the classification of the product descriptions. The models tested are logistic regression, linear Support Vector Classification (SVC), Naïve Bayes, Random Forest and XGBoost.

Preprocessing and feature engineering like vectorization, tokenization, removal of stop words and use of n-gram can impact the accuracy of the models results. However, we cannot say in advance which combination of these transformations will generate the best results.

With that in mind, we tried all the possible combinations of these preprocessing and feature engineering techniques associated to each model. The idea here is not to evaluate the impact of the preprocessing and feature engineering but have the best score among all combinations.

The gain in applying these preliminary data treatments is not a consensus among practitioners and the problem at hand as can be found in the different reports given by the authors of Refs. Kruczek-Szepel and Piatkowska [2020] beneficial) and Measure [2020].

The combinations of preprocessing and feature engineering tested for all models are described below:

- Vectorization: [Countvectorizer, tfidf]
- Tokenizer: [None, Stem, Lemma]
- Stopwords: [None, stopwords]
- n-gram: [unigram, bigram]

Different sets of hyperparameters were considered to derive an optimal result. We list below the options that were evaluated for each ML model.

1. Logistic regression:

   - C: [0.01, 0.1, 1, 10, 100]
   - l1_ratio: [0, 0.25, 0.5, 0.75, 1]

   The saga solver was adopted because it is faster for larger datasets, has multinomial approach and enables all penalizations, including elastic-net which is the combination of L1 and L2 penalizations.

2. Linear SVC:

   - C: [0.1, 1, 10]
   - penalty: [l1, l2]
   - loss: [hinge, squared_hinge]
   - dual: True
   - multi_class: ovr
   - random_state: 42, for reproducibility.

3. Stochastic Gradient Descent (SGD) classifier. Allows comparison between logistic regression and linear SVC at once:

   - alpha: [0.00000001, 0.0000001, 0.000001, 0.00001, 0.0001, 0.001]
   - l1_ratio: [0, 0.25, 0.5, 0.75, 1]
   - loss: [log, hinge, squared_hinge]. log is equivalent to Logistic Regression, hinge is equivalent to Linear SVC with hinge loss and squared_hinge is equivalent to Linear SVC with squared hinge loss.

4. Naïve Bayes classifier:

   - alpha: np.linspace(0, 1, 20)
   - fit_prior: [True, False]

5. Random Forest:

   - bootstrap: [True, False]
   - oob_score: [True, False]
   - criterion in [gini, entropy]
   - n_estimators: [50, 100, 150, 200]
   - max_features: [None, sqrt, log2]
   - min_samples_leaf: [1, 2, 3]
   - random_state was defined as 42, for reproducibility.

6. XGBoost:

   - max_depth': [8, 10, 12],
   - gamma: [0, 0.1]
   - eta: [0.01, 0.05, 0.1]

3.2 Model(s) finally selected and quality criteria used (e.g., accuracy, time)[which model was selected? What quality measures were used to compare different ML models (e.g. accuracy (e.g. RMSE, MAE, F1, precision), runtime to train the model (e.g. 2 hours for 500,000 training samples and 25 features))]

To evaluate the performance of the models studied we make use of standard metrics such as accuracy, precision, recall and f1-score.

The global result for these metrics is shown in Table 4. As can be seen the classification report for all models are quite similar. The best figures are highlighted in blue and show that the linear SVC and Logistic Regression presents the best results with a dominance of the former, with the Linear SVC displaying better accuracy and weighted average metrics.

Furthermore, table 5 shows that the Linear SVC has much faster hyperparameter tuning than Logistic Regression with an execution time almost as fast as Naive Bayes.

The simplest model, Naïve Bayes, is less accurate than the others, though runs in a faster pace, as shown in table 5, while the Random Forest is the slowest one.

Table 4: Classification report summary for all the studied models.

Model	Accuracy	Precision	Recall	F1_score	
Logistic Regression	0,9799	0,8980	0,8945	0,8908	macro avg
		0,9786	0,9799	0,9783	weighted avg
Linear SVC	0,9836	0,8888	0,8988	0,8900	macro avg
		0,9828	0,9836	0,9826	weighted avg
SGDC	0,9813	0,8916	0,8903	0,8868	macro avg
		0,9792	0,9813	0,9795	weighted avg
Naive Bayes	0,9513	0,8144	0,7634	0,7691	macro avg
		0,9513	0,9513	0,9471	weighted avg
Random Forest	0,9713	0,8625	0,8641	0,8553	macro avg
		0,9698	0,9713	0,9685	weighted avg
XGBoost	0,9680	0,7898	0,7942	0,7836	macro avg
		0,9651	0,9680	0,9650	weighted avg

Table 5: Comparison of execution time for hyperparameter tuning of the ML models.

Model	Execution Time
Logistic Regression	02h50m36s
Linear SVC	00h06m06s
SGDC	00h46m28s
Naive Bayes	00h04m17s
Random Forest	46h12m03s
XGBoost	14h11m30s

In agreement with previous studies for classification of CPI products Myaklatun [2019], Kruczek-Szepel and Piatkowska [2020] the results illustrated above also suggest that the Linear SVC is the best option for our case as well.

The combination of hyperparameters and data preparation techniques that led to the optimal results for the Linear SVC is given below:

- Lemma
- Stop words
- tf-idf
- 1-gram

- C=10
- penalty=l2
- loss=hinge

3.3 Hardware used

AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx, 2.30 GHz, RAM 12,0 GB

4 Results and discussion

In this section we discuss some additional results at a more granular level obtained for the linear SVC model. Since we want to apply the models to classify the products to a given category, the global results can be misleading in a sense that they do not tell us how the models are performing for each individual case and mask possible problems and limitations.

Table 6 shows the classification report for the linear SVC model for each individual subitem. Most of the categories present high scores over 90% of correct classification of the products belonging to it.

However, for some cases the performance is poor with some null scores. This is more prominent of fresh food categories whose sample sizes are very small (see the support column in Table 6, which counts the number of cases in the test sets for each category, for a more concrete illustration).

As an example, the first category in Table 6, abacate (avocado in English), contains only one observation in the test set. In fact, there is only one observation in the whole data set, hence no observation for this category in the training set. Due this, the model does not know how to attribute a given product to this subitem and will try to attribute to a different one which will also lead to a reduction in the classification of this other category and the global one as a by product.

Table 6: Classification report for the best model - Linear SVC

Subitem	Precision	Recall	F1-score	Support
Abacate	0,0000	0,0000	0,0000	1
Abobrinha	1,0000	1,0000	1,0000	2
Acém	1,0000	1,0000	1,0000	2
Alcatra	1,0000	1,0000	1,0000	7
Alface	1,0000	1,0000	1,0000	16
Alho	0,6667	1,0000	0,8000	2
Arroz	0,9910	1,0000	0,9955	110
Atomatado	1,0000	0,9878	0,9939	82
Atum em conserva	1,0000	1,0000	1,0000	20
Azeite de oliva	1,0000	1,0000	1,0000	56
Azeitona	1,0000	1,0000	1,0000	32
Açúcar cristal	1,0000	0,8333	0,9091	12
Açúcar demerara	0,0000	0,0000	0,0000	2
Açúcar refinado	0,7895	1,0000	0,8824	15
Bacalhau	1,0000	1,0000	1,0000	6
Balas	1,0000	1,0000	1,0000	37
Banana-da-terra	1,0000	1,0000	1,0000	2
Banana-maçã	1,0000	1,0000	1,0000	1
Banana-prata	1,0000	1,0000	1,0000	6
Batata-doce	0,8750	1,0000	0,9333	7
Batata-inglesa	1,0000	1,0000	1,0000	3
Biscoito	1,0000	0,9965	0,9982	286
Bolo	1,0000	1,0000	1,0000	37
Brócolis	1,0000	1,0000	1,0000	5

Continue on the next page

Table 6: Classification report for the best model - Linear SVC (cont.)

Subitem	Precision	Recall	F1-score	Support
Café moído	0,9811	0,9905	0,9858	105
Café solúvel	0,9688	0,9688	0,9688	32
Caldo concentrado	1,0000	1,0000	1,0000	51
Capa de filé	1,0000	1,0000	1,0000	1
Carne de porco	0,8846	0,9200	0,9020	25
Carne de porco salgada e defumada	0,9000	0,9000	0,9000	10
Carne-seca e de sol	1,0000	0,8571	0,9231	7
Cebola	0,8333	1,0000	0,9091	5
Cenoura	0,8750	1,0000	0,9333	7
Cerveja	1,0000	1,0000	1,0000	134
Cheiro-verde	0,7500	1,0000	0,8571	3
Chocolate e achocolatado em pó	0,8800	0,9565	0,9167	23
Chocolate em barra e bombom	0,9924	0,9848	0,9886	132
Chá mate (erva mate)	1,0000	1,0000	1,0000	30
Chã de dentro	1,0000	1,0000	1,0000	3
Coentro	1,0000	0,6667	0,8000	3
Contrafilé	1,0000	1,0000	1,0000	4
Costela	0,5000	0,6667	0,5714	3
Couve	0,5000	1,0000	0,6667	1
Couve-flor	0,0000	0,0000	0,0000	1
Creme de leite	1,0000	1,0000	1,0000	10
Farinha de arroz	1,0000	1,0000	1,0000	8
Farinha de mandioca	0,9783	1,0000	0,9890	45
Farinha de trigo	1,0000	1,0000	1,0000	31
Feijão-carioca (rajado)	1,0000	1,0000	1,0000	9
Feijão-macáçar (fradinho)	1,0000	1,0000	1,0000	1
Feijão-preto	1,0000	1,0000	1,0000	4
Fermento	1,0000	0,6667	0,8000	6
Filé-mignon	1,0000	1,0000	1,0000	9
Flocos de milho	0,5000	1,0000	0,6667	2
Frango em pedaços	0,9703	0,9800	0,9751	100
Frango inteiro	0,9167	0,8462	0,8800	13
Fubá de milho	1,0000	0,5000	0,6667	4
Fígado	1,0000	1,0000	1,0000	1
Goiaba	0,0000	0,0000	0,0000	0
Hambúrguer	1,0000	1,0000	1,0000	5
Inhame	1,0000	1,0000	1,0000	4
Iogurte e bebidas lácteas	0,9893	1,0000	0,9946	185
Lagarto comum	1,0000	1,0000	1,0000	4
Laranja-pera	0,7778	1,0000	0,8750	7
Leite condensado	1,0000	1,0000	1,0000	21
Leite de coco	1,0000	0,8333	0,9091	12
Leite em pó	0,9412	1,0000	0,9697	32
Leite fermentado	1,0000	0,9643	0,9818	28
Leite longa vida	0,9870	1,0000	0,9935	76
Limão	1,0000	1,0000	1,0000	6
Linguiça	1,0000	1,0000	1,0000	71
Macarrão	0,9852	0,9568	0,9708	139
Macarrão instantâneo	1,0000	1,0000	1,0000	52
Maionese	1,0000	1,0000	1,0000	31

Continue on the next page

Table 6: Classification report for the best model - Linear SVC (cont.)

Subitem	Precision	Recall	F1-score	Support
Mamão	1,0000	1,0000	1,0000	10
Mandioca (aipim)	0,8000	0,8000	0,8000	5
Manga	1,0000	1,0000	1,0000	7
Manteiga	1,0000	1,0000	1,0000	32
Maracujá	0,0000	0,0000	0,0000	1
Margarina	1,0000	1,0000	1,0000	32
Massa semipreparada	0,9490	0,9789	0,9637	95
Maçã	1,0000	1,0000	1,0000	10
Melancia	1,0000	1,0000	1,0000	5
Melão	1,0000	1,0000	1,0000	3
Milho (em grão)	1,0000	1,0000	1,0000	8
Milho-verde em conserva	1,0000	1,0000	1,0000	17
Molho de soja	1,0000	1,0000	1,0000	5
Morango	1,0000	1,0000	1,0000	3
Mortadela	1,0000	1,0000	1,0000	17
Músculo	1,0000	1,0000	1,0000	4
Outras bebidas alcoólicas	1,0000	0,9643	0,9818	28
Ovo de galinha	1,0000	1,0000	1,0000	7
Palmito em conserva	1,0000	1,0000	1,0000	7
Patinho	1,0000	1,0000	1,0000	3
Peixe-atum	1,0000	1,0000	1,0000	2
Peixe-cação	0,0000	0,0000	0,0000	1
Peixe-pescada	0,0000	0,0000	0,0000	1
Peixe-salmão	1,0000	1,0000	1,0000	6
Peixe-sardinha	0,6667	1,0000	0,8000	2
Peixe-tambaqui	0,0000	0,0000	0,0000	1
Peixe-tilápia	1,0000	1,0000	1,0000	5
Pepino	1,0000	1,0000	1,0000	1
Pera	0,0000	0,0000	0,0000	2
Picanha	1,0000	1,0000	1,0000	4
Pimentão	1,0000	1,0000	1,0000	3
Polpa de fruta (congelada)	1,0000	1,0000	1,0000	23
Presunto	1,0000	1,0000	1,0000	14
Pá	0,0000	0,0000	0,0000	1
Pão de forma	0,9859	0,9211	0,9524	76
Pão de queijo	1,0000	1,0000	1,0000	17
Pão doce	0,9091	0,9375	0,9231	32
Pão francês	1,0000	1,0000	1,0000	5
Queijo	0,9932	1,0000	0,9966	146
Refrigerante e água mineral	1,0000	1,0000	1,0000	130
Repolho	1,0000	1,0000	1,0000	5
Requeijão	0,9545	1,0000	0,9767	21
Sal	1,0000	1,0000	1,0000	13
Salsicha	1,0000	1,0000	1,0000	23
Sardinha em conserva	1,0000	1,0000	1,0000	19
Sopa desidratada	1,0000	0,8889	0,9412	18
Sorvete	1,0000	1,0000	1,0000	107
Suco de frutas	0,9912	1,0000	0,9956	113
Suco em pó	1,0000	1,0000	1,0000	60
Tempero misto	0,9894	0,9789	0,9841	95

Continue on the next page

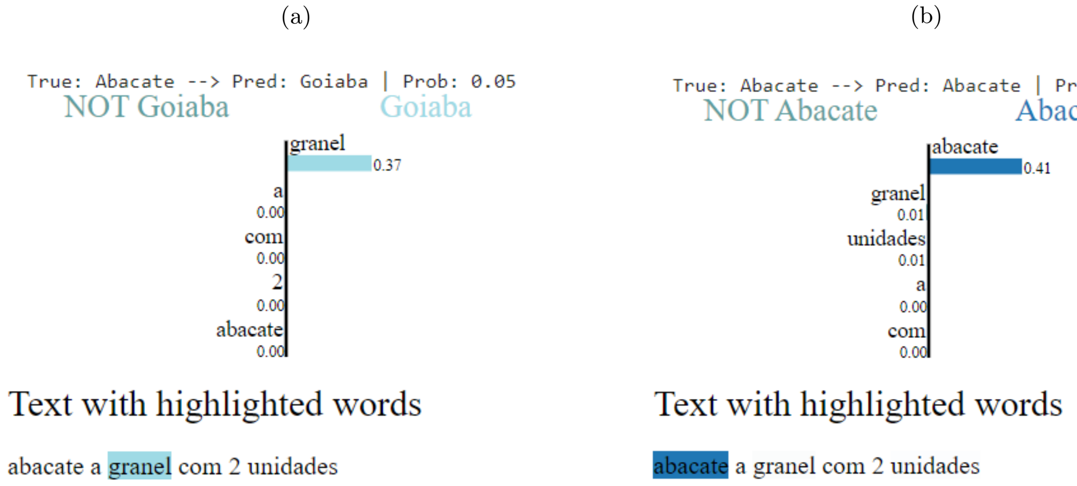
Table 6: Classification report for the best model - Linear SVC (cont.)

Subitem	Precision	Recall	F1-score	Support
Tomate	1,0000	1,0000	1,0000	5
Uva	1,0000	1,0000	1,0000	7
Vinagre	1,0000	1,0000	1,0000	7
Vinho	0,9853	1,0000	0,9926	67
Óleo de soja	1,0000	1,0000	1,0000	14

For the case of the abacate, the model assigned the observation to goiaba (guava in English). An interesting tool to understand further why this choice was made is the LIME library [Ribeiro et al., 2016]. LIME stands for Local Interpretable Model-agnostic Explanations and is a method for explaining predictions of ML models [Ribeiro et al., 2016].

Figure 3a displays the LIME's output which explains why the product description *abacate a granel com 2 unidades* was attributed to Goiaba. The model does not understand "abacate" as an important word for the category Abacate but attributes a great weight to the word "granel" to ascertain it to Goiaba.

A more straightforward way to improve the results is simply to add more product descriptions for those scarce categories. For instance, in Figure 3b we show LIME's output for the same product description after the insertion of two other abacate's descriptions from another data set. After running the model with this addition, it is now able to predict the product to the correct category and to identify "abacate" as the most important word for this. Not only the assignment is correct but also the probability of classification increased in comparison with the last misclassification using the previous data set.



(a)

True: Abacate --> Pred: Goiaba | Prob: 0.05

Text with highlighted words

abacate a granel com 2 unidades

(b)

True: Abacate --> Pred: Abacate | Prob: 0.17

Text with highlighted words

abacate a granel com 2 unidades

Figure 3: (a) LIME's output illustrating why the description *abacate a granel com 2 unidades* was attributed to the Goiaba subitem. The most important word considered was granel. (b) LIME's output after adding additional products belonging to the "abacate" subitem. Now the model was able to correctly classify the product and identifies abacate as the most important word as expected.

Pera (pear in English) category is a similar case as abacate. It has two observations for this category in the whole data set, though two of them in the test set and none in the training set. Furthermore, there is a category called laranja-pera (orange-pear in English) which has a few observations in the training set. It is interesting to note here that in addition to the lack of training of the model, the word pera (most important for classifying a product to Pera) is also very representative for the classification of a product into Laranja-Pera as shown in the LIME's output displayed in Figure 4.

Another interesting case is the one displayed when the model tries to classify the product "pão de mandioca sabor caseiro 450g", a bread made with manioc. It should be classified into the Pão de Forma class, which is a class with a large sample of products. However, the model attributes this to the Mandioca (manioc) subitem.

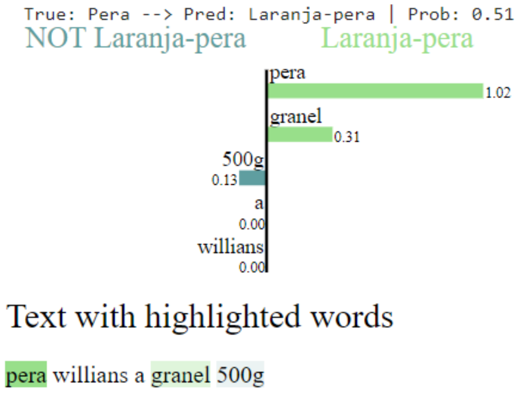


Figure 4: LIME's output illustrating why the description *Pera Williams a granel 500g* was wrongly attributed to the Laranja-pera subitem. The word pera, the most important for the Pera subitem is also very relevant for the Laranja-Pera one.

As Figure 5 shows in this case the model identifies the word pão (bread in english) as an important word to assign for one of the pão categories. However, the model prioritizes the word mandioca to assign the description for the Mandioca (aipim) category.

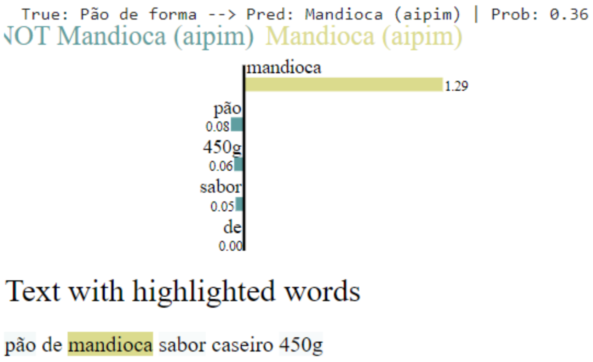


Figure 5: LIME's output illustrating why the description *pão de mandioca sabor caseiro 450g* was attributed to the mandioca subitem instead of the correct one pão de forma. The common word mandioca has a very strong weight towards the mandioca subitem.

Besides the more obvious approach of introducing more data to improve the performance of the models, this last example illustrates that even for categories with suitable training samples, some manual work to check possible inconsistencies is necessary. Authors in Refs. Myaklatun [2019], Kruczek-Szepel and Piatkowska [2020] make use of the probability ascertained by the models to classify a given product to derive criteria for manual inspection of the results of the automatic classification. Of course a trade-off between the level of accuracy and manual labour should also be considered here as presented in Ref. Myaklatun [2019].

Improvement of our training samples, for instance via combining data from the CPI sample and derivation of criteria for inspection of the data obtained are some of the steps we shall take in the continuity of this project.

5 Code/programming language

The codes are in Python using the functionalities provided by the scikit-learn, xgboost, lime, nltk, spacy, pandas and numpy libraries.

6 Evolution of this study inside the organisation [e.g. Has this study advanced ML within the organisation? Was there any collaboration within the organisation?]

For the development of this study there was no collaboration with other members of the institution.

But we believe that, beyond ourselves, this study has greatly promoted the advancement of ML to other colleagues from the institution. Under the CPI department we have promoted discussions in regular meetings with other colleagues in which we provide reports on the updates of this work.

We have also disseminated the activities of the UNECE ML group with other colleagues during this year beyond our own project. We invited other members of the institution either from the prices' department or other sectors to take part in the activities of the UNECE group as listeners (workstream and general meetings, extra seminars and workshops, coffee and coding etc) so that other members of the institution could be aware of the rich discussions in different areas being conducted in this forum. In our opinion these actions have also served to disseminate the interest in this area across the institution.

7 Is it a proof of concept or is it already used in production? [If it is a proof of concept: Was it successful? How will its results prospectively be used in the future?]

This is still a proof of concept. We consider that the proof of concept to be successful and very promising though additional tests and developments are still necessary to evaluate the robustness of the methods.

The results will be used as part of undergoing research for further use of alternative data sources for the CPI produced at IBGE.

7.1 What is now doable which was not doable before?

Classification and aggregation of large volume of products into CPI are challenging and key tasks. With this study we made our first steps to get further insight into these problems and devised preliminary tools to deal with the classification part of the problem.

The study considered only basket categories related to groceries, but we can expand the study to other categories in a straightforward manner. Once we can have access to other data sources such as scanner data, we shall be faced with similar problems and the machinery being developed for the classification of web data would be applied to this problem as well. In this sense we are also gaining time in the future to incorporate other data sources with the knowledge gained and the solutions already being constructed.

The codes implemented can probably be adapted for other NLP classification tasks at IBGE allowing the optimization of tasks currently performed manually.

7.2 Is there already a roadmap/service journey available how to implement this?

There is not a roadmap designed yet, but important steps to consider in this way are: additional tests on the models with other sectors of the CPI basket, how the models classify data as new products are incorporated along time, construction of a platform for manual assessment of the results to validate and improve the models, evaluation of changes in the IT system and in the production team routines to incorporate such changes.

7.3 Who are the stakeholders?

So far, the stakeholder is IBGE's price index division. We consider that other divisions of IBGE could be benefited by this study. For instance, the sector responsible for the conduction of Household Budget Surveys also needs to classify data on expenditures in goods and services into a classification system of their own. This also relies on attributing products descriptions to a given category. An extension of the methods conducted here to this scenario should be straightforward.

7.4 Fall Back

As we are still in a research phase of this project, no fall back plan is implemented. However, among important points to consider one might include the definition of more sensitive sectors to select for a manual labelling in case of failure of the automatic approach.

7.5 Robustness

At this moment the tests were performed for data sets obtained for a given collection period. In order to incorporate such techniques into production we need to check the robustness of the models to consistently classify products along time but also its ability to correctly classify new products that will rise due market dynamics. Manual checks will probably be necessary to evaluate how accurate are the results being provided in this process.

8 Conclusions and lessons learned [e.g. ML can be used for editing but one has to have the following points in mind ...]

Taking part in the activities of the ML group was a very enriching experience for ourselves since we were able to take part in a great number of discussions reporting a variety of studies on the use of machine learning for statistical purposes.

In our opinion the existence of shareable and reproducible material of the projects is another of the strengths of this community. This allows new users to learn ML tools faster via following the steps of a given applied problem. Regarding this point, for our case with almost null experience with ML or NLP tools, the available material provided by Stats Poland Kruczek-Szepel and Piatkowska [2020] on a similar project has given a great support for us to speed up the development of our own codes. We thank these authors for the hard work to release a rich arxiv with codes, data sets and reports.

The study developed shows that use of ML for automatic classification of products for CPI is very promising. This is a key point for making greater use of alternative data sources for the CPIs produced at IBGE.

Additional tests are still necessary to evaluate the robustness of the models against other sectors of the CPI baskets and how the models classify new products as we add new data sets with products extracted at different periods of time.

The results obtained via use of web data can also be applied to classify products from other big data sources such as scanner data and help to speed up the construction of a system able to integrate use of different kinds of big data sources into the CPI routines.

9 Potential organisation risk if ML solution not implemented

The study is still ongoing, but if a ML solution for automatic classification is not adopted, the incorporation of alternative data sources for CPI purposes will probably be impacted since their use will probably be restricted and will not account the full potential of these sources.

10 Has there been collaboration with other statistical organisations, universities, etc?

The web data used in the last two quarters were extracted by scrapers developed by researchers from the Federal University of Minas Gerais (UFMG) in a project of collaboration with IBGE.

11 Next steps

We intend to keep participating in the activities of the ML group in the next round and keep developing our study in parallel.

We intend to refine the results presented here initially via addition of more data to train the categories with poor performance due data scarcity. According to the results of this approach we aim to look at more sophisticated techniques to deal with the problem of classifying the classes with few observations.

We also want to evaluate how the models deal with classification of products from other sectors of the CPI basket and how they perform on new products obtained for different collection periods. The problem of defining groups of homogeneous products whose mean price is tracked along time instead of a single product should also be assessed for categories such as clothing which is characterized by strong data churn and attrition.

The development of criteria and a tool for manual validation of the data automatic classified are other points that desire attention in our next steps. A nice example in this area is the study presented in Ref. Myaklatun [2019] which uses a combination of criteria based on the probability of allocation of the products in a given category and the relative difference between the categories ranked in the top two positions for a given product.

There is also an ocean of ML and NLP techniques available today and we aim to learn more about other techniques such as the word2vec and Bert models which are more powerful and sophisticated tools than the ones we studied here. This will help us in the development of other collaborations (which are running in parallel to this project) in which such techniques are already being adopted.

Appendices

Here we provide some of the results obtained during this year for other data sets.

A First quarter results

This set is based on web scraped data for a reduced set of products belonging to electronics and household appliances subitems.

The distribution of products according the different categories is shown in Figure 6.

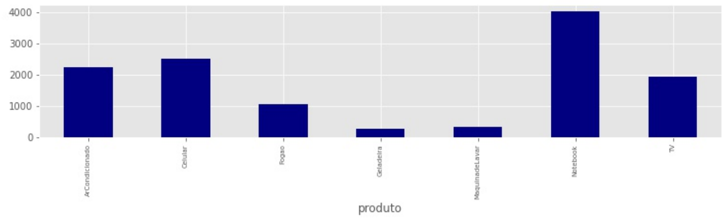


Figure 6: Number of descriptions by categories for the first quarter data - electronics and home appliances.

After use of logistic regression with Countvectorizer the model was able to classify all data sets without any flaw.

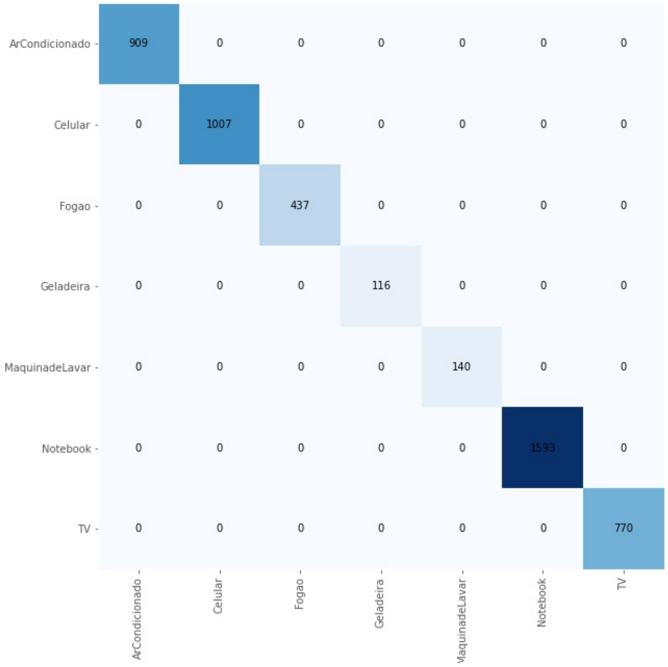


Figure 7: Confusion matrix using logistic regression model for the first quarter data - electronics and home appliances. The model correctly assigns all product descriptions to the respective category.

B Second quarter results

In the second quarter and part of the third quarter we considered a data set with products descriptions obtained from the CPI frame of products. This set is more robust as encompasses a larger number of categories.

Figure 8 displays the distribution of products for different categories.

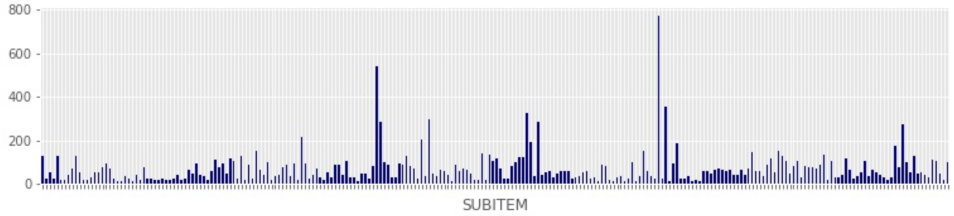


Figure 8: Distribution of product descriptions for the second quarter data.

Table 7 displays the classification report for this data set. As can be seen, the models have a very high global performance.

Table 7: Classification report summary for the best model - linear SVC

Model	**Accuracy**	**Precision**	**Recall**	**F1_score**	
Logistic Regression	0,9676	0,9731	0,9632	0,9665	macro avg
		0,9690	0,9676	0,9672	weighted avg
Linear SVC	0,9693	0,,9725	0,9663	0,9679	macro avg
		0,9705	0,9693	0,9690	weighted avg

References

A. G. Chessa. Mars: A method for defining products and linking barcodes of item relaunches. *Paper presented at the 16th meeting of the Ottawa Group*, May 2019. URL https://eventos.fgv.br/sites/eventos.fgv.br/files/arquivos/u161/product_definition_with_mars_chessa_og19.pdf.

L. T. da Silva, I. L. de Oliveira, T. Dantas, and V. G. Miranda. Studies of new data sources and techniques to improve cpi compilation in brazil. *Paper presented at the 16th meeting of the Ottawa Group*, May 2019. URL https://eventos.fgv.br/sites/eventos.fgv.br/files/arquivos/u161/study_of_new_data_sources_snipc_lincoln_da_silva.pdf.

Eurostat. *Practical guidelines on web scraping for the HICP.* Eurostat, November 2020. URL https://ec.europa.eu/eurostat/documents/272892/12032198/Guidelines-web-scraping-HICP-11-2020.pdf.

A. Harms and S. Spinder. A comprehensive view of machine learning techniques for cpi production. *CBS discussion paper series*, Nov 2019.

M. Honnibal and I. Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 2017.

M. Kruczek-Szepel and K. Piatkowska. Ecoicop classification. *Report of the project developed under the UNECE working group on ML in the 2020 round.*, 2020.

E. Loper and S. Bird. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028, 2002. URL http://dblp.uni-trier.de/db/journals/corr/corr0205.html#cs-CL-0205028.

H. Martindale, E. Rowland, and T. Flower. Machine learning for classification with big data in price statistics production pipelines. *Paper presented at the 16th meeting of the Ottawa Group*, May 2019. URL https://eventos.fgv.br/sites/eventos.fgv.br/files/arquivos/u161/semi-supervised_ml_for_price_stats-ottawa_group.pdf.

M. Measure. Automatic classification of work-related injury and illness narratives. *Report of the project developed under the UNECE working group on ML in the 2020 round.*, 2020.

V. G. Miranda, L. T. da Silva, A. F. G. Almeida, and P. K. da Costa. Measuring inflation of ride sharing services in brazilian official cpis. *Paper presented at the online ILO/UNECE meeting of the Group of Experts on Consumer Price Indices*, June 2021. URL https://unece.org/sites/default/files/2021-05/Session_4_Brazil_Paper.pdf.

K. H. Myaklatun. Utilizing machine learning in the consumer price index. *Paper presented at the Nordic Statistical Meeting NSM2019*, Aug 2019.

I. Oliveira, J. Cesario, R. Molina, and V. G. Miranda. Using web scraping tools in collecting traveller accommodation prices to improve cpi compilation in brazil. *Work presented at the Eurostat Workshop on Scanner Data and Web Scraping*, October 2021. URL https://circabc.europa.eu/ui/group/7b031f10-ac19-4da3-a36f-58708a70133d/library/e9fff896-7c7c-4e5b-a57d-9c5ab3effd8f/details?download=true.

ONS. *Classification of new data in UK consumer price statistics.* April 2021. URL https://www.ons.gov.uk/economy/inflationandpriceindices/articles/classificationofnewdatainukconsumerpricestatistics/2021-04-06.

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

M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.