Agricultural Exports Classification

Project

270957 – Machine Learning

Students

Sergio Postigo

Víctor Diví

Professor

Bernat Coma

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

June, 2022

Table of Contents

1 Introduction

1.1 Problem to work on

Obtaining categories for imported/exported goods based on the information registered by the customs office of the respective country (Peru).

1.2 Reason/motivation

Many countries base a big portion of their economies in foreign trade. Therefore, the customs agencies around the world collect data about every imported/exported good that passed across their ports, airports, borders, etc. In some countries, this data is open, and anyone can access to it to analyze it and make more informed decisions while importing or exporting goods. However, this data demands some challenges before its use. One of them is the labelling. For example, in Peru every time a good is imported/exported, a customs agent fills a form with the information of the product(s), where they include descriptions about it. Nonetheless, there isn´t a proper labelling that can help for example, to aggregate amounts imported/exported by category.

1.3 Data available

The data was provided from a consultancy company in Peru called Aurum[1]. They were hired by an agricultural exports company that was interested in knowing which categories of agricultural products were exported from Peru from 2017 till 2021. Aurum acquired the data from the company Veritrade[2], who consolidate foreign trade databases from many countries in South America. The data includes 41 columns, with textual, numerical, and categorical values. The number of rows is 631394. A summary of all the columns is presented below.

[1] https://aurumperu.com/
[2] https://www.veritradecorp.com/en

Column	Description
Partida Aduanera	Specific code of a product included in the Harmonized System of the World Customs Organization (WCO)
Descripcion de la Partida Aduanera	Description about the product associated with the customs code
Aduana	Customs office from which the export was performed
DUA	Single Administrative Document, it is a document that gathers information about the shipping
Fecha	Shipping date
Año	Shipping year
Cod. Tributario	Tax code of the company exporting the good
Exportador en Perú	Company or entity exporting the good
Importador Extranjero	Company or entity importing the good
Kg Bruto	Weight of the good in kg, including the weight of the container or box
Kg Neto	Weight of good in kg, excluding the weight of the container or box
Toneladas Netas	Weight of good in tons, excluding the weight of the container or box
Qty 1	Quantity of the good in terms of a specific measurement unit (1)
Und 1	Unit of measurement (1)
Qty 2	Quantity of the good in terms of a specific measurement unit (2)
Und 2	Unit of measurement (2)
U$ FOB Tot	The value of the goods at the exporter's customs frontier in USD
Miles de USD Fob TOTAL	The value of the goods at the exporter's customs frontier in thousands of USD
U$ FOB Tot	The value of the goods at the exporter's customs frontier in USD
U$ FOB Und 1	The value of the goods by unity (1)
U$ FOB Und 2	The value of the goods by unity (2)
Pais de Destino	Country of destiny
Puerto de destino	Port of destiny
Último Puerto Embarque	Last port of shipment
Via	Via (air, see, maritime)
Agente Portuario	Port agent
Agente de Aduana	Customs agent
Descripcion Comercial	Commercial description of the good
Descripcion1	Commercial description portion 1
Descripcion2	Commercial description portion 2
Descripcion3	Commercial description portion 3
Descripcion4	Commercial description portion 4
Descripcion5	Commercial description portion 5
Naviera	Shipping company
Agente Carga(Origen)	Load Agent (origin)
Agente Carga(Destino)	Load Agent (destiny)

Canal	Selectivity channel. Type of control that the Customs Service will carry out on the merchandise to be exported. There are three channels: Green, Orange and Red
Concatenar	Column that concatenates 27, 28, 29, 30, 31, 32
Categoría macro Aurum	Designated category/label
Subcategoría inicial	Designated subcategories/sub-lables
Subcategoría Consolidada Aurum	Designated subcategories/sub-lables (with less granularity, it groups some sub-categories in "others")
Categoría Consolidada Aurum	Designated category/label (with less granularity, it groups some categories in "others")

Table 1.1: Column details

1.4 Related previous work

The labelling for goods is done manually mainly by consultancy agencies, who get this data to generate analytics reports for companies and institutions interested in foreign trade information of specific products. They usually use MS Excel spreadsheets to perform the labeling, which is not efficient and takes much time.

1.5 Implementation

All code used for this project can be found in the attached zip file or available in the following Github repository: https://github.com/sergiopostigo/ML_Project (data files present in zip do not exist in the repository for their large size). To run the code, first install the dependencies listed in the Pipfile (requirements.txt is also provided). All code should be reproducible as the random states of all random functions are defined.

2 Data Exploration

In this section we present a summary of the whole data exploration performed in the notebooks. All the steps presented here are summaries of the ones implemented and further detailed in the scripts.

2.1 Data exploratory Analysis

This section is developed and detailed in the notebook *data_analysis.ipynb.* The main remarks are presented below.

2.1.1 Remark 1

All posible categories all labeled in *Categoria macro Aurum* and all possible subcategories are labelled in *Subcategoria inicial*. Aurum grouped some categories in *Categoria consolidada Aurum* as "others" and did the same in *Subcategoría Consolidada Aurum* for the subcategories. This last two columns were very likely a requirement from their client. He may have been interested specially in a list of categories and the rest were simply labeled as "others". However, what is from interest from us are the columns with all the categories and all the subcategories (*Categoria macro Aurum* and *Subcategoria inicial*).

Let's visualize the distribution of the categories:

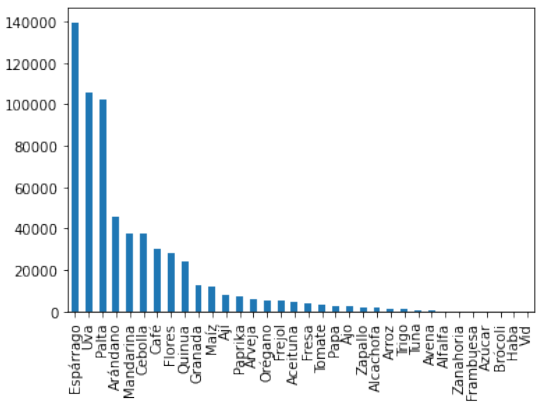

Figure 2.1: Histogram for Categoría macro Aurum

For the subcategories, there is also a big class imbalance. However, since there are a lot classes (more than 600) in this column, we show only the first 1000 rows distribution to show the idea:

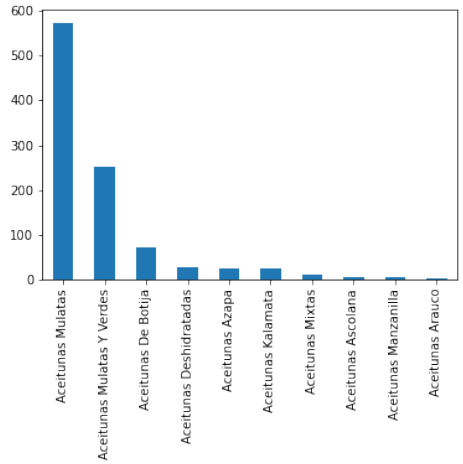

Figure 2.2: Histogram for Subcategoria inicial

We can see that both columns have a big imbalance, and that the subcategory column has a lot of labels with very few rows. Most of these subcategories come from varieties of the same product and combinations of these varieties. For example, see the number of different subcategories that are subcategories of blueberries:

```
Arándanos Frescos Sin Variedad                                                    40881
Arándanos Congelados Sin Variedad                                                 1395
Arándanos Frescos Rojos                                                           882
Arándanos Frescos Ventura                                                         564
Arándanos Varios Sin Mayor Detalle                                                332
                                                                                  ...
Arándanos Frescos Biloxi - Ventura - Emerald - Snowchaser - Stella Blue           1
Arándanos Frescos Biloxi - Ventura - Emerald - Springhigh                         1
Arándanos Frescos Bb-01 - Bb-03 - Bb-04                                           1
Arándanos Frescos Biloxi - Atlas - Bb-02 - Bb-05 - Bb-06                          1
Arándanos Frescos Biloxi - Emerald - Corrina - Dupree - Kirra - Snowchaser - Terrapin - Stella Blue    1
```

Since this is likely to happen with more categories, we will keep *Categoría macro Aurum* as the target class we want to predict. Besides, having 34 classes is more than enough, if we chose the subcategory, we would be dealing with more than 600 classes, and obtaining a decent classifier would be unrealistic.

2.1.2 REMARK 2

The columns *Descripcion1*, *Descripcion2*, *Descripcion3*, *Descripcion4* and *Descripcion5* concatenated build *Descripcion Comercial.* Additionally, we can make an additional remark

here: *Descripcion Comercial* has repeated sentences in its values, as is showed in the example below, likely due to different description fields containing the same information. This will have to be cleaned.

2.1.3 Remark 3

The column *Concatenar* concatenates *Descripcion Comercial* and *Descripcion1*, *2,3,4* and *5*. Thus, it basically has a concatenation of two times the string from *Descripcion Comercial*. It seems that the consultants didn't know that *Descripcion1*, *2,3,4* and *5* are trims of *Descripcion Comercial*. Maybe they thought this extra column contained additional information and that is why they decided to concatenate everything in the *Concatenar* column to then process the information from here.

2.1.4 Remark 4

For each *Partida aduanera* there is only one possible *Descripcion de la partida aduanera*. The number of all combinations of the columns *Partida Aduanera* and *Descripcion de la Partida Aduanera* are 220. The number of unique values of the column *Partida Aduanera* is 220. The number of unique values of the column *Descripcion de la Partida Aduanera* is 201. There are some values of *Descripcion de la partida aduanera* that correspond to multiple values of *Partida Aduanera.*

	Partida Aduanera	Descripcion de la Partida Aduanera
4879	603199000	LAS DEMÁS FLORES Y CAPULLOS, CORTADOS PARA RAM...
127592	603129000	LAS DEMÁS FLORES Y CAPULLOS, CORTADOS PARA RAM...
128022	603149000	LAS DEMÁS FLORES Y CAPULLOS, CORTADOS PARA RAM...
58226	713609000	LAS DEMÁS HORTALIZAS (INCLUSO SILVESTRES) DE V...
601093	713349000	LAS DEMÁS HORTALIZAS (INCLUSO SILVESTRES) DE V...
93615	713359000	LAS DEMÁS HORTALIZAS (INCLUSO SILVESTRES) DE V...
3871	904229000	LAS DEMÁS PIMIENTA DEL GÉNERO PIPER; FRUTOS DE...
585226	904211090	LAS DEMÁS PIMIENTA DEL GÉNERO PIPER; FRUTOS DE...
584597	904219000	LAS DEMÁS PIMIENTA DEL GÉNERO PIPER; FRUTOS DE...
286679	1008509000	LOS DEMÁS ALFORFÓN, MIJO Y ALPISTE; LOS DEMÁS ...
286910	1008109000	LOS DEMÁS ALFORFÓN, MIJO Y ALPISTE; LOS DEMÁS ...
433452	1008909100	LOS DEMÁS CEREALES
286689	1008902900	LOS DEMÁS CEREALES
286733	1008909900	LOS DEMÁS CEREALES
...		

Table 2.1: Same Customs codes with multiple descriptions

2.1.5 Remark 5

Since we are dealing with agricultural items, we can presume that there is a seasonal influence in the dates in which they are exported. Let's test this assumption by selecting 3 random values of *Partida Aduanera* and plotting the count of shipments of those products

across time. The random selected values from *Partida Aduanera* are: [1212920000, 805210000, 713399100] and their plots:

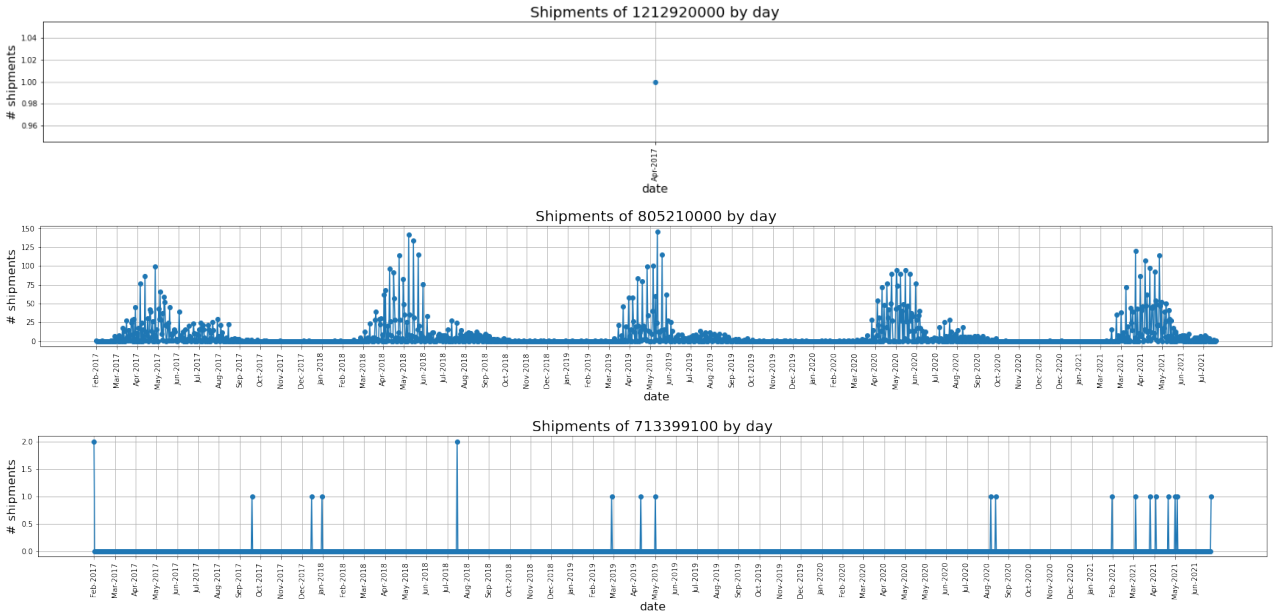

Figure 2.3: Count of shipments across time for different products

As we can see, some products (represented by its *Partida Aduanera* number) present a seasonal pattern (as expected) but others not.

2.1.6 Remark 6

The column *Descripcion de la Partida Aduanera* gives general information about the associated product code of Partida aduanera, while the column Descripcion Comercial gives information about the specific shipment. There are 98606 different values of *Descripcion Comercial* in total

2.2 Data cleaning

In this stage we will clean the data and specifically the columns that we will use in the model(s) in the next section. This is developed and detailed in the notebook *data_cleaning.ipynb*. Of course, we don´t need to clean all the columns, since many of them

are not relevant for labeling the rows. So, let's determine the columns to be used and justify why[3] and specify the cleaning needed.

Column	Type	Justification	Cleaning
Descripcion de la Partida Aduanera	Textual	This is a general description about the product, so this carries valuable information for the labeling.	Remove stop words, accents, punctuations, and non alphabetic characters. Set to lowercase
Fecha	Categorical	Associating the date of shipping to a category is insightful. As we saw, some products are exported in specific seasons of the year.	Get the month (numerical format) and remove anything else.
Kg Neto	Numerical	The weight of the shipments is insightful, but is highly variable among same products, so initially we won´t use this feature. However, we will use it to calculate the price by kg, which is insightful.	Get the price by kg of the good. To do this we will use both columns and transform them into one. The new column's name is *usd_kg*
U$ FOB Tot	Numerical	The cost of the shipment will be used to calculate the cost by kg of the product	
Pais de Destino	Categorical	The country where these products are being exported can be related to groups of products	Remove accents and set to lowercase.
Descripcion Comercial	Textual	The commercial description also carries valuable information for the labeling	Remove accents, double or more white spaces, stop words, punctuations and set to lowercase.
Categoría macro Aurum	-	LABEL	-

Table 2.2: Selected columns and cleaning

2.3 Data preprocessing

In this stage we will preprocess the data to be used in the classification models. In the code this is presented in the script *data_preprocessing.ipynb.* As seen in the Data Exploration section, there is a big class imbalance. We will address this issue as first step.

[3] The columns not being used and the corresponding justification can be found in the script *data_cleaning.ipynb*

The strategy taken here down-sampling to reduce the number of instances of the more popular categories. For each category we will have at most 20.000 instances. After this the categories remain as shown below:

Palta	20000
Mandarina	20000
Uva	20000
Quinua	20000
Arándano	20000
Café	20000
Cebolla	20000
Espárrago	20000
Flores	20000
Granada	12672
Maíz	11946
Ají	7814
Paprika	7178
Arveja	6140
Orégano	4997
Frejol	4993
Aceituna	4890
Fresa	3975
Tomate	3151
Papa	2699
Ajo	2228
Zapallo	2117
Alcachofa	1553
Arroz	1308
Trigo	957
...	
Azúcar	22
Brócoli	10
Haba	2
Vid	2

Figure 2.4: Categories count after down-sampling

Until now, we have worked over the whole dataset since the actions performed would be also done over new data. However, the next steps should only be performed with the training data, so we will split the data into two sets (80-20) and carry on working with only the 80% of the data.

So far, we know that we are dealing with text, categorical and numerical data in this dataset. The next step will be then to represent the text columns as vectors of numbers, which is known as *sentence embedding*. This will be done in the columns *Descripcion de la Partida Aduanera* and *Descripcion Comercial* and it's crucial to them in the models. In this step we use the Gensim library and specifically the Doc2Vec function. The chosen vector size is 10, as we will show in the modeling section, it leads to good results.

3 Data Modeling

In this section we introduce the models used in this project. The notebook where this is developed is *models.ipynb*

3.1 Modeling methods considered

Type	Model	Option	Features used
Non-Linear	Multi-Layer Perceptron	1	• *Descripcion Comercial*
		2	• *Descripcion de la Partida Aduanera*
		3	• *Descripcion Comercial* • *Descripcion de la Partida Aduanera*
		4	• *Descripcion Comercial* • *Descripcion de la Partida Aduanera* • *Fecha* • *Usd_kg* • *Pais de Destino*
Linear	Ridge Regression	5	• *Descripcion Comercial* • *Descripcion de la Partida Aduanera* • *Fecha* • *Usd_kg* • *Pais de Destino*
	Logistic Regression	6	• *Descripcion Comercial* • *Descripcion de la Partida Aduanera* • *Fecha* • *Usd_kg* • *Pais de Destino*
Ensembles	Random Forest	7	• *Descripcion Comercial* • *Descripcion de la Partida Aduanera* • *Fecha* • *Usd_kg* • *Pais de Destino*
	Gradient Boosting	8	• *Descripcion Comercial* • *Descripcion de la Partida Aduanera* • *Fecha* • *Usd_kg* • *Pais de Destino*
	Gradient Boosting (Light GBM)	9	• *Descripcion Comercial* • *Descripcion de la Partida Aduanera* • *Fecha* • *Usd_kg* • *Pais de Destino*

Table 3.1: Modeling methods and the features used

3.2 Validation Protocol

We study the accuracy and F-1 score of the predictions compared to the testing data. We are using both scores to have a clear view of how different classes are predicted. Since we have unbalanced data, considering only the accuracy we could end up with models that predict very well overrepresented classes while mispredicting underrepresented ones. This way F1-socre gives us an idea of the general classification capability of the model among all the classes.

Additionally, we include confusion matrices for every option, as shown in *models.ipynb*. After this first run, we select the two best performant models and then we do an optimization of their parameters using grid search and cross-validation.

4 Results

The results are summarized in the following table. The two selected models for grid search are shown in green. We select the best performant MLP and the Random Forest.

Option	Accuracy	F-1 score	Parameters
1	40.90%	18.86%	hidden_layer_sizes = (16, 8, 8)
2	88.91%	63.30%	hidden_layer_sizes = (16, 8, 8)
3	89.42%	62.77%	hidden_layer_sizes = (16, 8, 8)
4	91.15%	64.63%	hidden_layer_sizes = (16, 8, 8)
5	73.04%	37.29%	Default
6	44.43%	18.99%	Default
7	96.93%	80.53%	Default
8	93.78%	70.72%	Default
9	72.59%	43.22%	Default

Table 4.1: Performance of every option

Then, we get the best performant set of parameters for these two models and its corresponding accuracy and F-1 score results (these values are the averages from different cross-validations). The best results shown in green below and the configuration used in the first run in blue.

Hidden layers	alpha	Time (s)	Accuracy	F-1 score
8-4	1e-3	78.0	76.7%	39.5%
8-4	5e-4	66.0	75.9%	39.2%
8-4	1e-4	78.0	76.8%	39.7%
16-8-8	1e-3	102.0	90%	63.2%
16-8-8	5e-4	102.0	89.7%	62.3%
16-8-8	1e-4	84.0	89.7%	62.4%

Table 4.2: Parameter optimization results for Multi-Layer Perceptron

For Multi-Layer Perceptron we get slightly better results increasing the alpha a bit. Using the testing data, the accuracy is 89.32% and F1-score is 61.27%. We can mention that the performance with testing data is slightly worse.

max_depth	n_estimators	Time (s)	Accuracy	F-1 score
10	50	12.05	86.6%	56.7%
10	100	22.7	86.8%	56.7%
10	200	44.9	87.0%	57.0%
50	50	19.7	96.0%	76.7%
50	100	38.2	96.3%	77.5%
50	200	81.0	96.4%	77.8%
inf	50	19.3	96.0%	76.7%
inf	100	38.2	96.3%	77.5%
inf	200	81.0	96.4%	77.8%

Table 4.3: Parameter optimization results for Random Forest

For Random Forest we get better results increasing the max_depth and number of estimators. Using the testing data, we get an accuracy of 96.96% and F-1 score 80.58%. Note that the results with max_depth of 50 and no max_depth are the same, this is because all the created trees have less than 50 levels.

While the Gradient Boost didn´t performed nearly as well as other methods in the first run, we decided to explore a bit more on it because we wanted to explain why its result was so different from Random Forest. As we can see in the table below, with more appropriate parameters it gives good results. With these parameters, it gets an accuracy of 95.99% and an F1-score of 80.87% with the final test data.

learning_rate	n_estimators	Time (s)	Accuracy	F-1 score
0.5	50	3.9	24.7%	8.9%
0.5	100	6.1	24.7%	8.9%
0.5	200	11.3	24.7%	8.9%
0.1	50	7.1	82.5%	54.6%
0.1	100	13.2	31.3%	14.0%
0.1	200	17.9	31.3%	14.0%
0.01	50	6.6	92.5%	72.9%
0.01	100	12.8	94.1%	74.8%
0.01	200	25.7	95.8%	76.6%
0.001	50	6.2	72.4%	28.8%
0.001	100	11.9	86.9%	67.0%
0.001	200	23.1	90.0%	72.8%

Table 4.4: Parameter optimization results for Gradient Boosting (Light GBM)

4.1 Final model chosen

After the comparisons performed in the previous section, we can conclude that the best model is:

Model	Features used	Parameters	Accuracy	F-1 score
Random Forest	• *Descripcion Comercial* • *Descripcion de la Partida Aduanera* • *Fecha* • *Usd_kg* • *Pais de Destino*	Max_depth: inf N_estimators: 200	96.96%	80.58%

Table 5.5: Best model

5 Conclusions and future work

In this project we tested different predictive models to label row with 34 possible categories of agricultural products exported from Peru in a certain period. Our approach was to first make a fast fit using different models with their default parameters to have an idea of which of them would perform better. Once we had the first results, we chose the best performing models and carried out an optimization process to get the best parameters for them. The two best models were Multi-Layer Perceptron and Random Forest, this last one being not only more accurate but also significantly faster to train.

Given the good results, we see feasible to use the models obtained here in a real scenario as a support for human manual classification (or even completely substitution). However, there are some aspects that should be considered before applying these methods in a production environment. For example, these methods are focused only on Peruvian agricultural data, so it's possible that they would not perform as well in the same type of data from other countries.

A very significant next step for this work would be to find a model to classify subcategories. As it was mentioned, the main problem was the big number of subcategories (more than 600) existing in the data, which may impact the performance of classifiers. However, now that we can us as input the predicted "categories" column, the labeling of subcategories looks way more realistic, since this attribute would be very insightful for such a model.

Additionally, it would be interesting to explore ways of creating interfaces that consume the model(s) created in this project and predict the categories for an "non-tech" end user. A Jupyter Notebook is always a useful tool for data scientists, but not intuitive for any consultants or people interested in foreign trade data. It is always important to think in the final user.

Finally, as mentioned before, we would like to extrapolate the approach in this project into foreign trade data of different topics. It would be interesting to assess the performance of our models trained in non-agricultural data.