Automatic Product Classification in International Trade: Machine Learning and Large Language Models*

Ignacio Marra de Artiñano
Université Libre de Bruxelles and ECARES

Franco Riottini Depetris
Inter-American Development Bank and UdeSA

Christian Volpe Martincus
Inter-American Development Bank and CESifo

This version: August 2024

Abstract

Accurate product classification is crucial in international trade. In this study, we apply and assess several algorithms to automatically classify agricultural and food products based on text descriptions sourced from different public agencies, including customs authorities and the United States Department of Agriculture (USDA). We find that while traditional machine learning (ML) models tend to perform well within the dataset in which they are trained on, their precision drops dramatically when applied to external datasets. In contrast, large language models (LLMs) show a consistently strong performance across all datasets. The top performing LLMs —Claude 3.5 Sonnet and GPT 4— achieve accuracy rates of approximately 80% at classifying products into 6-digit Harmonized System (HS) categories and above 90% for HS 2-digit Chapters. Our analysis highlights the valuable role that artificial intelligence can play in facilitating product classification at scale and, more generally, in enhancing the categorization of unstructured data.

Keywords: Product Classification, Machine Learning, Large Language Models, Trade
JEL Codes: F10, C55, C81, C88

*We would like to thank Peter Schott for valuable comments and suggestions. We are also grateful to Victoria Patience for her careful edition and María Lidia Viquez for her excellent assistance in the publication process. The views and interpretations in this paper are strictly those of the authors and should not be attributed to the Inter-American Development Bank, its executive directors, or its member countries. Ignacio Marra de Artiñano gratefully acknowledges funding from the Fund for Scientific Research - FNRS (FC 57977).
Contact: Christian Volpe Martincus: christianv@iadb.org.

1 Introduction

Accurately classifying products is essential in international trade. Virtually all countries use the Harmonized System (HS) nomenclature to categorize products into tariff lines for both statistical and duty collection purposes. Misclassification, both intentional and unintentional, can be very costly. It can result in imprecise measurement of trade flows, inappropriate determination of origin, foregone duty collection, and significant delays in border monitoring and processing. Furthermore, it can lead to the design and implementation of misguided trade policies, especially those related to trade remedies such as countervailing duties, antidumping, and safeguards.

Traditionally, the bulk of product categorization tasks has been carried out manually, frequently based on experts' judgments, and has accordingly been extremely time-consuming.[1] As a consequence, classification is challenging for governments, firms, and researchers, especially on a large scale. This has been magnified by the rise of cross-border e-commerce, which requires customs agencies to process several million small shipments per year. In many developing countries, this has generally resulted in a large share of shipments being classified based on their value or size instead of the specific goods they consist of, thus limiting their ability to conduct risk assessments properly and that of their countries to accurately measure the composition of a growing portion of their trade. Firms, in turn, particularly those that are small or have no previous experience exporting or importing, typically find it difficult to assign their products to HS codes and need to rely on costly specialized services to do so.[2] Last but certainly not least, many databases contain product-level information in the form of unstructured product names or text descriptions. This makes it hard for researchers to combine them, leading to time-intensive and imperfect merges with standard trade databases based on the HS nomenclature.

[1] As highlighted by public customs' agencies resolutions, classification is often the object of firms' ex-ante consultations and is subject to ex-post adjustments.

[2] Furthermore, in an effort to reduce the wrong attribution of tariff lines, custom agencies often impose heavy misreporting fines, which can be burdensome for exporters.

The advent of machine learning (ML) is likely to reduce the cost of these classification efforts and increase their accuracy (see WCO, 2022a).[3] While there is an incipient literature that aims to assess the precision of ML for product classification, most existing efforts rely on tests on the same dataset used to train them. As a consequence, there is very limited evidence on how these models perform on real external datasets and, hence, on their general applicability. Further, such an evidence is missing altogether in the case of large language models (LLMs), which are yet to be tested at scale for this purpose.

In this paper, we examine the performance of a variety of ML models along with LLMs -GPT 3.5, GPT 4, Claude 3 Sonnet and Claude 3.5 Sonnet-, at classifying products into the HS nomenclature.[4] We assess its performance at different aggregation levels, including the 2-digit HS Chapter level (HS2), the 4-digit HS Heading level (HS4) and the 6-digit HS Product Subheading level (HS6). For instance, the HS6 code 080510 refers to "Oranges", which belongs to the HS4 Heading 0805 "Citrus fruit" and which in turn is part of the HS2 Chapter 08 "Edible fruits and nuts".

When classifying product descriptions into HS codes we will go beyond the train-and-test dataset and thus explicitly assess the external validity of the models. For this, we will use three different datasets: (i) a dataset containing product descriptions from the Chilean customs agency to train and test ML algorithms, thus following earlier literature; (ii) a dataset containing product descriptions from the customs agency of a different country, Paraguay; and (iii) a database of product descriptions from the United States Department of Agriculture (USDA).[5] This third data source describes products for which firms obtain an organic certification. In all cases, our analysis will be limited to animal, vegetable, and food products since these are the product categories for which firms can obtain organic certification (Marra

[3]The BACUDA project run by the World Customs Organization (WCO) is an example of ongoing work using these techniques for customs applications.

[4]We acknowledge that there are several other possible approaches to automatic product classification, including convoluted neural networks (CNN), recurrent neural networks (RNN), and other transformer-based ML models. In this paper, we restrict ourselves to some of the most widely used and practitioner-friendly ML and LLMs.

[5]We use Paraguayan customs data because, like Chilean data, they are publicly available.

de Artiñano et al., 2024).[6]

Our results reveal that, while standard ML algorithms performed very well within the train-and-test set, their accuracy dropped dramatically when these models were applied to datasets on which they were not explicitly trained. In contrast, LLMs performed very evenly across all datasets.

Their accuracy was very high: the two top performing models (Claude Sonnet 3.5 and GPT 4) achived accuracy rates of 73-88% at the HS 6-digit level, 81-93% at the HS 4-digit level, and 89-95% at the HS 2-digit level across the 3 datasets. Overall, we find that the models with larger parameter size and training sets perform better, that is, GPT 4 on average performs better than GPT 3.5 and Claude Sonnet 3.5 performs better than Claude Sonnet 3.[7]

There are several important applications for this sort of scalable automatic product classification that uses product descriptions as inputs. First, it could help customs agencies identify mistmatches between reported product descriptions and HS codes and thus detect patterns of intentional or fraudulent product categorization.[8] Second, it would make it easier for both policymakers and researchers to categorize product descriptions from unstructured data sources (such as those obtained from e-commerce transactions, bank statements, electronic billing machines or from historical sources) into established product nomenclatures. Finally, it could be used to develop chatbots that give HS code suggestions from simple text descriptions, which would greatly facilitate tariff line attribution for firms engaged in international trade and even consumers participating in cross-border e-commerce.[9]

We make two main contributions to the existing literature. First, we show that the accuracy and overall performance of ML algorithms for automatic product classificaiton is high

[6]This project was originally conceived with the objective to match product descriptions of organic certified firms with their corresponding HS Codes. In a future study, we will extend the analysis to all HS products.

[7]The parameters in LLMs are the number of variables that a model adjusts during training, typically the weights within neural network layers. A larger parameter space thus implies a higher capacity to adjust to a variety of linguistic patterns and subtleties and, hence, to react to different prompts.

[8]Some firms may try to avoid paying higher duties by misrepresenting a product as another relatively similar product with a lower tariff. LLMs could thus be used to flag mismatches between reported product descriptions and reported HS codes, potentially triggering a customs inspection.

[9]The US Census has already developed a similar chatbot; see https://uscensus.prod.3ceonline.com/.

within the test-train dataset, but declines sharply when such algorithms are applied to external datasets. Second, this study is, to the best of our knowledge, the first to apply large language model LLMs at scale to the WCO's HS product classification and, more generally, to a large-scale multiclass classification problem assigning standardized categories to sectoral or product text descriptions.[10]

A number of previous studies have proposed alternative approaches to automatically classify products into HS codes across many tariff lines. Spichakova and Haav (2020) use ML methods to provide 6-digit HS code predictions and recommendations using a model trained with product descriptions from the US Bill of Lading 2017 database.They show that the algorithm achieves an accuracy of 80% on the test dataset. Ruder (2020) uses a variety of ML and deep learning models to classify product descriptions from the US Bill of Lading and reaches accuracy levels of approximately 60%. Chen et al. (2021) apply unsupervised ML and an off-the-shelf embedding encoder to automatically assess whether reported HS codes in cross-border import declarations are correct. They achieve an overall success rate of 71% on an HS 6-digit dataset provided by Dutch customs. Turhan et al. (2015) adopt a different strategy whereby they use visual properties along with product labels and descriptions. The accuracy level they achieve is above 80% with 4-digit HS codes from a database of 4,494 binding tariffs published by the European Union in 2014.

These papers use a single dataset, which is split into training and testing samples. Unfortunately, the use of a single dataset prevents these authors from testing the accuracy of the models in external datasets. This limitation is crucial because tariff databases often have significantly different product descriptions and text formats. One exception in this regard is He et al. (2021), who use data gathered directly from firms to train their models, along with a second dataset of product descriptions from a third firm that was not in the test dataset. However, they focus on very few HS products (12 6-digit potential product classifications)

[10]Kocon et al. (2023) carry out a simpler classification exercise focused on only a few categories. We also tested the performance of GPT 3.5 in mapping sector descriptions onto the North American Industry Classification System (NAICS). To do so, we used sectors reported by firms when registering with the online business platform *ConnectAmericas*.

and their exercise is accordingly much simpler than product categorization across the universe of potential tariff lines.

We contribute to this literature on automatic product classification by assessing the accuracy of different ML algorithms on both the test-train-split dataset and two additional datasets for a large set of products. Our results indicate a very large decrease in the accuracy of standard ML algorithms outside the dataset on which the models are trained.

There is also recent literature that aims to apply GPT and other LLM models to text-based data in the social sciences. Some recent papers that use GPT include Hansen et al. (2023), Lopez-Lira and Tang (2023), Hansen and Kazinnik (2023), Yang and Menczer (2023), and Ko and Lee (2023).[11] Hansen et al. (2023) compare the performance of a predecessor of GPT-3 to their own model, WHAM, and find that WHAM outperforms GPT-3 in terms of the error rate at the task of classifying whether a job posting allowed the possibility of remote work at least one day per week. The authors also discuss the potential gains of adopting modern natural language processing (NLP) methods for text classification in economic environments. They suggest that other prediction problems using text in economics might similarly benefit from a large training sample combined with sequence embedding models, such as GPT-3.

Lopez-Lira and Tang (2023) examine the potential of OpenAI's GPT 3.5 in predicting stock market returns by using analysis and the classification of news with potential impact for firms. Their analysis suggests that, even though GPT 3.5 is not specifically trained for this task, it produces superior results in terms of predicting stock market returns than other traditional sentiment analysis methods commonly used in finance due to the comprehensiveness of the model. In a similar vein, Ko and Lee (2023) show that GPT 3.5 effectively helps improve portfolio management by selecting asset classes that statistically outperform random choices in diversification and returns. Hansen and Kazinnik (2023) use GPT 3.5 and GPT 4 to decipher Fedspeak, the language used by the Federal Reserve to communicate monetary policy decisions. Their results suggest that these models obtain the lowest numerical

[11] An exhaustive analysis of the recent literature using GPT (and its adjacent models) is beyond the scope of this paper. Nevertheless, it is worth mentioning papers such as Noy and Zhang (2023) on the effects on productivity, Biswas (2023) on its potential role in health, and Kasneci et al. (2023) on its potential impact on education.

errors, the highest accuracy rates, and the highest measure of agreement relative to human classification when compared to other pretrained linguistic models and dictionary-based approaches. Finally, Yang and Menczer (2023) use GPT 3.5 to study the credibility of news and conclude that they are able to correctly evaluate news sources by rating them.

We add to these papers by showing the usefulness of LLMs for product classification in international trade. We find that while the top-performing LLMs (Claude Sonnet 3.5 and GPT 4) perform slightly worse than traditional ML algorithms on the test-train-split dataset, they significantly outperform these models on external databases. The reason is that LLMs are able to go beyond the specific context of the training dataset and thus have much higher external validity. Unlike traditional ML algorithms, they also require no additional data-cleaning or preprocessing, making them much simpler to use.

The rest of this paper is structured as follows: Section 2 describes the different data sources used in our analysis. Section 3 explains the methodological approach. Section 4 discusses the main automatic classification results and explores a series of extensions and robustness checks. Finally, Section 5 concludes with a brief discussion of our results.

2 Data

In this paper, we used three different freely available datasets: a database of product descriptions from Chilean customs (National Customs Service of Chile, 2023), a database of product descriptions from Paraguayan customs (National Customs Agency of Paraguay, 2023), and a database of organic product descriptions from the USDA (USDA Integrity Dataset, 2023).[12] Customs transaction records include both the product description (as provided and recorded by the exporter/importer) and HS-codes (self-reported by the exporter/importer).[13] A product description typically contains detailed information about the goods in a shipment, which may include the nature of the goods, their composition, and/or their intended use. The level

[12] Table A1 in the Online Appendix provides the URLs where these data can be obtained.

[13] Importantly, firms can receive hefty fines for incorrectly describing the product and/or attributing the wrong HS codes, regardless of whether such misattribution is intentional or not.

of detail and specific information included can vary across different product descriptions.[14] The first database (Chilean customs) was used to train and test the ML algorithms. The second database (Paraguayan customs) was employed to test the external validity of our models. Finally, the third database (USDA) was used to further test the models outside the context of customs product descriptions.

Our goal is to classify products into the Harmonized System, by far the most widely classification used by customs agencies worldwide.[15] The entire Harmonized System has 5,612 codes at the HS-6 level, 1,222 at the HS-4 level, and 97 codes at the HS-2 level. More specifically, we aim to map descriptions to HS codes related to agricultural and food products, which correspond to HS-2 Chapters 1 to 22. Thus, for the purpose of this paper the number of potential classes is 866 HS-6 codes, 185 HS-4 codes and 22 HS-2 codes.

2.1 Train-Test-Split Dataset: Trade Transactions from the Chilean Customs

To generate and train the ML models predicting the HS nomenclature for target products, we used Chilean export and import transactions from 2009 to 2021 as our train-and-test dataset. This comprehensive dataset encompasses over 104 million observations with granular information on trade transactions, including detailed HS codes and product descriptions.

We focused on HS Chapters 1–22, covering agricultural, animal, and food products. To manage computational load, we randomly selected 1 million product descriptions from these HS Chapters.[16] Note that our sample of 1 million product descriptions corresponds to unique descriptions; that is, we delete duplicated descriptions before our randomization.[17] Following standard practice in the ML literature, we used 70% of this sample for training purposes

[14] Thus, a specific HS code has many different correct descriptions according to the description provided by the exporter/importer.

[15] Most countries around the world -including Chile and Paraguay — share the Harmonized System product nomenclature up to 6 digits. More granular classification systems (8-digits HS, 10-digits HS) tend to be country-specific.

[16] We randomize over the universe of product descriptions in Chapters 1-22 in Chilean customs. Importantly, this implies that our data is not balanced across HS-2 Chapters. In Online Appendix A10, we balance the data by HS 2- Chapters, finding similar results to those in our baseline unbalanced models. See Section 4.4 and Online Appendix A10 for more information.

[17] Online Appendix A4 show the code for this randomization process.

and the remaining 30% for testing.

2.2 External Dataset 1: Trade Transactions from the Paraguayan Customs

To test our algorithms against a dataset outside the training set, we used a random sample of product descriptions from trade transactions recorded by Paraguayan customs. As before, we restricted the sample to agricultural, animal, and food products (HS Chapters 1–22). Importantly, this dataset includes not only the product descriptions but also the HS codes assigned by firms, thus enabling us to directly test the accuracy of the HS codes provided by the different ML algorithms, by GPT models and by Claude models. [18]

2.3 External Dataset 2: USDA Organic Product Descriptions

Finally, we used information on products for which the USDA has issued organic certifications to Latin American firms (see Marra de Artiñano et al., 2024). The original dataset comprises more than 26,000 product descriptions. These texts vary substantially in terms of how specific and clean they are; that is, whether they use clear, easy-to-understand wording that is narrow enough to accurately categorize the product. Thus, these descriptions may be significantly shorter than those usually found in customs databases (e.g., "maize" or "mangoes"), and may be highly specific or scant (e.g., "concentrate soursop pulp" or "ungurahui"). Online Appendix A1 shows a random set of 10 product descriptions for illustrative purposes. In this case, the original database does not include the HS6 codes. We thus manually classify by hand the HS6 codes in order to be able to test the accuracy of the different models in this dataset.

[18] In a robustness check, we use these trade transactions from Paraguay as the train-test-split dataset and the Chilean transactions as the first external dataset. We find very similar results; see Section 4.4. and Online Appendix A11.

3 Methodology

Classification algorithms play a vital role in a wide range of ML applications (Sarker, 2021).[19] Multiclass classification, one of the most common applications of classification algorithms, presents a particular challenge. The goal is to categorize data into three or more distinct and mutually exclusive categories (Aly, 2005). This process involves training one or more models to accurately assign uncategorized data to the correct categories. Formally, given a training dataset of the form (x_i, y_i) where x_i is the ith input and y_i is the ith class label that belongs to the set $\{3, \ldots, N\}$ we want to find a model H such that $H(x_i) = y_i$ for new, uncategorized data.

The process of automatic product classification using ML models consists of several steps. First, the train-and-test data -in our case, the product descriptions in trade transactions obtained from Chilean customs- needs to be preprocessed, including tasks such as preliminary cleaning, tokenization and feature extraction. Second, the data must be divided into the training and testing sets. Third, a series of different ML algorithms are applied to the training set. After performing these steps, we also tested the estimated models on two alternative external databases (product descriptions in trade transactions from Paraguayan customs and the USDA organic product database).

Our analysis was entirely conducted using Jupyter notebooks and Python open-source libraries such as NLTK, scikit-learn, spaCy, AST, and other commonly used libraries. In addition, we use the OpenAI (GPT) and Anthropic (Claude) APIs to classify the different products through direct prompts and benchmark its performance against that of the ML models.

[19] They have been used extensively in areas such as NLP (Otter et al., 2020), image recognition (Fujiyoshi et al., 2019; Lai, 2019), and sentiment analysis (Mitra, 2020), among others domains. In recent years, breakthroughs in NLP and text mining have propelled the adoption of these algorithms in applications as diverse as fraud detection, asset classification in finance, and early detection of health problems (Kowsari et al., 2019).

3.1 Data Processing

As mentioned above, the Chilean customs dataset covers 2009–2021 and contains more than 104 million observations. We processed this dataset by first restricting the product descriptions to those in chapters 1–22 of the HS schedule, which correspond to animal, vegetable, and food manufacturing products. This first filter reduced the total number of observations to approximately 12 million and the number of unique 6-digit HS codes to 866. [20] We then proceeded to randomly select 1 million product descriptions in an effort to reduce the computational burden of the exercise.

To clean and preprocess the product descriptions, we performed a series of tasks, which are summarized in Table 1:

[20] In addition, we filter out 469,435 observations that do not correspond to any known product according to the standard HS nomenclature (e.g., 160000).

Table 1: Preprocessing of Product Descriptions

Step	Description	Main Packages
Text preparation	We imported the Natural Language Toolkit (NLTK) library and applied the "word tokenize" function to break the text into individual words (tokens). This was crucial, as it made post-processing of text and feature extraction easier.	`ast, literaleval`
Lowercase	We converted all words to lowercase using a lowercase function. This helped to ensure that words are treated consistently in subsequent steps and to reduce data complexity.	`lower`
Normalization of non-ASCII characters	We applied a function to normalize non-ASCII characters, except for words that have the letter "ñ"[21]. This allows us to standardize and simplify the text, thus facilitating subsequent analysis.	`unicodedata, normalize`
Converting numbers written in words to digits	We used a function from the NLTK package to convert numbers written in words to digits. This helped reduce the complexity of the text and made it easier to extract relevant features.	`w2n, word_to_num`
Stop-word removal	We used a function to remove stop-words that do not provide relevant information for analysis, such as prepositions and conjunctions. This helped reduce the complexity of the text and allowed us to work on the most significant words.	`stopwords`
Lemmatization	The lemmatize functions were used to transform words into their base or lemma form. This helped reduce the complexity of the text by grouping similar words together and made it easier to identify patterns in the data.[22]	`nltk, spacy`
Removing words that are not in English or Spanish	We applied a function to remove words that are not in English or Spanish. This helped focus the analysis on the relevant languages and reduced noise in the data.	`langdetect, nltk`
English and Spanish noise removal	We applied some functions to remove irrelevant words in English and Spanish. This helped reduce noise in the data and allowed the most relevant words to be used for analysis.[23]	

Source: Authors' own elaboration.

[21] In those cases, we leave the word completely.

[22] For example, if we have different sentences that use the words "roasted," "roasting," and "roast," the lemmatization process will unify these words to their lemma, "roast". It is important to note that lemmatization does more than just remove inflectional endings. It performs a morphological analysis of each word.

[23] For English descriptions, we remove common logistical terms (e.g., 'invoice', 'cargo', 'shipment'), conjuctions and articles (e.g., 'and', 'of', 'the'), color names (e.g., 'red', 'blue', 'white'), and other frequently occurring non-informative terms (e.g., 'code', 'certify', 'order', 'number'). Similarly, for Spanish descriptions, we eliminate terms related to logistics (e.g., 'factura', 'carga', 'envío'), conjunctions and articles (e.g., 'y', 'de', 'el'), color names (e.g., 'rojo', 'verde', 'blanco'), and other non-informative terms (e.g., 'código', 'certificar', 'orden', 'número').

By cleaning and preprocessing the text in the product descriptions as described in these steps, we thus prepare the data for the ML modes, ensuring that they are as accurate and efficient as possible at estimating HS codes. Table A3 in the Online Appendix illustrates an example of this procedure. This example provides a clear idea of the complexity of dealing with certain descriptions and demonstrates the importance of the text-cleaning routine in traditional ML algorithms.

3.2 Traditional ML Algorithms

We used several different ML models for our multiclass classification problem. While offering an extensive explanation of such models is beyond the scope of this paper, this section contains a brief review of some of their characteristics, based primarily on Kowsari et al. (2019) and Aggarwal and Zhai (2012):

1. **Support Vector Machine (SVM)**: SVM is a supervised learning algorithm that identifies the optimal hyperplane that separates data points into their respective classes and maximizes the margin between the classes. The key in this classifier is to "determine the optimal boundaries between the different classes and use them for the purposes of classification" (Aggarwal and Zhai, 2012).

2. **Rocchio**: This algorithm operates by representing documents as vectors in a high-dimensional space. It calculates a centroid (average vector) for each category based on the training data. When classifying a new product description, the algorithm computes its vector representation and measures its similarity to each category centroid, assigning the description to the category with the closest centroid.

3. **Logistic Regression**: It is a linear model for binary classification, which can be extended to multiclass classification problems. Using a logistic function, the model estimates the probability of a product description belonging to a specific class. The class with the highest probability is then assigned to the product description.

4. **k-Nearest Neighbors (k-NN)**: It searches for the k most similar or closest items to the new object we want to classify, and then decides which category it belongs to, based on the most common category among its nearest neighbors.

5. **Random Forest**: It is an ensemble learning method that constructs multiple decision trees during training and combines their predictions to improve classification accuracy.

6. **Naive Bayes**: It is a probabilistic classifier based on Bayes' theorem, which assumes independence between features. Although this assumption is often not valid in real-world applications, Naive Bayes classifiers still perform well in many cases.

7. **Decision Tree**: It is a flowchart-like structure that can be used for classification tasks. The tree is built by recursively splitting the dataset based on the feature that provides the best separation into classes.

3.3 LLMs: GPT and Claude Sonnet

GPT 3.5, GPT 4, Claude Sonnet 3 and Claude Sonnet 3.5 are advanced large-scale language deep learning models. GPT 3.5 and GPT 4 were developed by OpenAI, while Claude was developed by Anthropic. In our analysis, we use the following OpenAI models: (1) GPT 3.5, which OpenAI refers to as "gpt-3.5-turbo" in their API documentation and which powered ChatGPT until May 2024, (2) GPT 4, which OpenAI designates as "gpt-4" in their API, and which has an estimated parameter size 10 times that of GPT 3.5. In addition, we use the following Anthropic models: (3) Claude 3 Sonnet and (4) Claude 3.5 Sonnet. Claude 3.5 Sonnet is -as of August 2024- the latest Anthropic model and Claude 3 Sonnet is its predecessor. They use transformer architecture to understand and generate human-like text. With billions of parameters and the ability to learn from vast amounts of text data, they have been fine-tuned to excel in a wide range of NLP tasks.

Some of the notable properties of these LLMs include their autoregressive nature, which allows them to generate contextually relevant and coherent text by predicting the next word in a sequence given the previous words. The models are trained using unsupervised learning

with vast datasets that include websites, books, and articles. The respective knowledge cut-offs for the LLM models used in this paper are September 2021 for GPT 3.5, December 2023 for GPT 4, August 2023 for Claude Sonnet 3 and April 2024 for Claude Sonnet 3.5. [24]

These models have been extensively benchmarked in recent literature. For instance, Zheng et al. (2023) evaluated GPT 4, GPT 3.5, and Claude V1 alongside other models using MT-bench (Multi-Turn Bench), a dataset consisting of 80 multi-turn questions designed to assess models' performance in extended dialogues and complex instruction following. They found that GPT-4's judgments closely aligned with human experts, achieving an 85% agreement rate, which was even higher than the agreement among humans (81%). Additionally, these models have been used as judges to evaluate other LLMs, with GPT-4 showing particularly strong performance in this role.

Recent literature has documented the rapid advancements in these LLM models. Korinek (2024) reports that as of May 2024, GPT-4o, an updated version of GPT-4, was widely regarded as the most capable publicly available LLM. The same study notes that Claude 3, introduced in March 2024, demonstrated exceptional proficiency in writing tasks. Korinek (2024) also highlights that these models have been subjected to a variety of benchmarks, including MMLU (Massive Multitask Language Understanding), a test that evaluates models' knowledge and reasoning across multiple disciplines; TruthfulQA, which measures models' ability to provide truthful information; and the aforementioned MT-bench. These evaluations have showcased the models' adaptability and high performance across a diverse range of tasks, further solidifying their position at the forefront of NLP technology (Korinek, 2024).

We applied the LLMs through the OpenAI and Anthropic APIs. For all models, we give a system command to act as a wizard that assigns 6-digit HS codes and then ask them to execute such function for a given product description. In this regard, it is worth mentioning that we ask the models not only to assign each product an HS code but also to provide its best estimate if the product description is not clear enough, thereby "forcing" it to make a

[24] Note that this is the knowledge cut-off as of the August 2024, when the last exercises in this paper were carried out. The models may be updated in the future. See https://platform.openai.com/docs/models/ and https://docs.anthropic.com/en/docs/about-claude/models for the latest knowledge cut-offs.

guess.

Cleaning the product descriptions before prompting the LLMs (that is, carrying out the data processing tasks described in section 3.1) is not necessary. When working with LLMs, which are trained on a diverse range of text typologies, preprocessing data may not be needed and may even be disadvantageous as it might obscure valuable contextual information. We therefore merely input orders one at a time, thus allowing all LLMs to categorize products individually. The specific prompts used, and the completion requests associated with them are presented in the Online Appendix (section A5).

4 Results

In this section, we present the main results of our analysis on automatic product classification using traditional ML algorithms and LLMs. Our evaluation is structured in three parts, corresponding to the three datasets used: results on the train-test split dataset (Chilean customs trade transactions), results on the first external dataset (Paraguayan customs trade transactions), and results on the second external dataset (USDA organic product descriptions). For each part, we analyse the accuracy of the models and other performance metrics, comparing traditional ML algorithms with LLMs. Our results reveal a clear pattern: while traditional ML algorithms perform well on the training dataset, their performance drops significantly when applied to external datasets. In contrast, LLMs show consistently strong performance across all datasets, with Claude 3.5 Sonnet and GPT-4 achieving the highest accuracy rates in all tests conducted.

4.1 Results on the Train-Test-Split Dataset: Trade Transactions from Chilean Customs

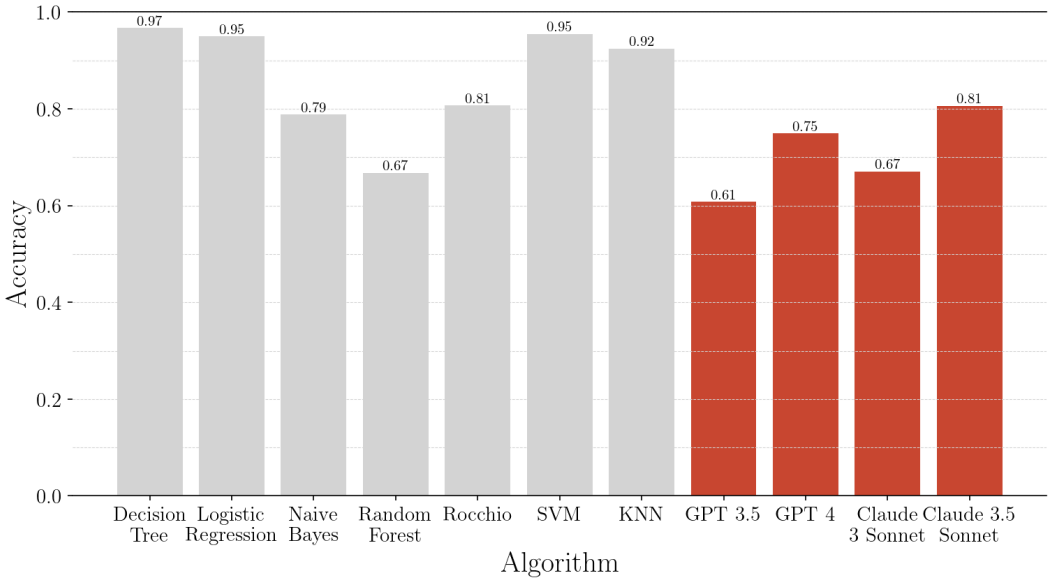
Model Accuracy. We firstly focus on the accuracy of the models, which refers to the share of correctly predicted HS product codes. Figure 1 shows the accuracy of the different models on the Chilean customs data. It is important to stress that this is the dataset on which the ML algorithms were trained. Note that GPT 3.5, GPT 4, and Claude are not "trained" using any

of the datasets since the outcomes are obtained from direct prompts to the models through their respective APIs.

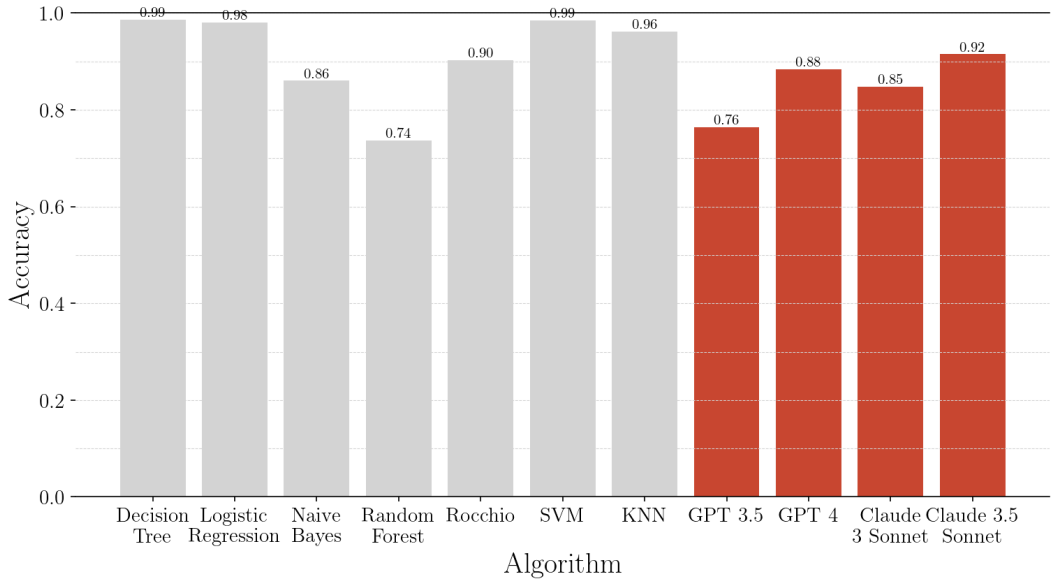
The trained ML algorithms have very high accuracy levels on this train-test dataset. The best performing algorithms are Decision Tree, Logistic Regression, and SVM (see Figure 1a). As expected, their accuracy increases when trying to classify products into less granular categories (see Figures 1b and 1c).

The top performing LLM is Claude 3.5 Sonnet, which achieves an accuracy rate of 81% at the HS 6 digits, 92% at 4 digits and 95% at 2 digits. It matches or outperforms some of the algorithms (including Naives Bayes, Random Forest and Rocchio). It is slightly below some others including Decision Tree and Logistic Regression. The second best performing model is GPT 4 (achieving 75% accuracy at HS 6), followed by Claude 3 Sonnet (68%) and GPT 3.5 (61%).

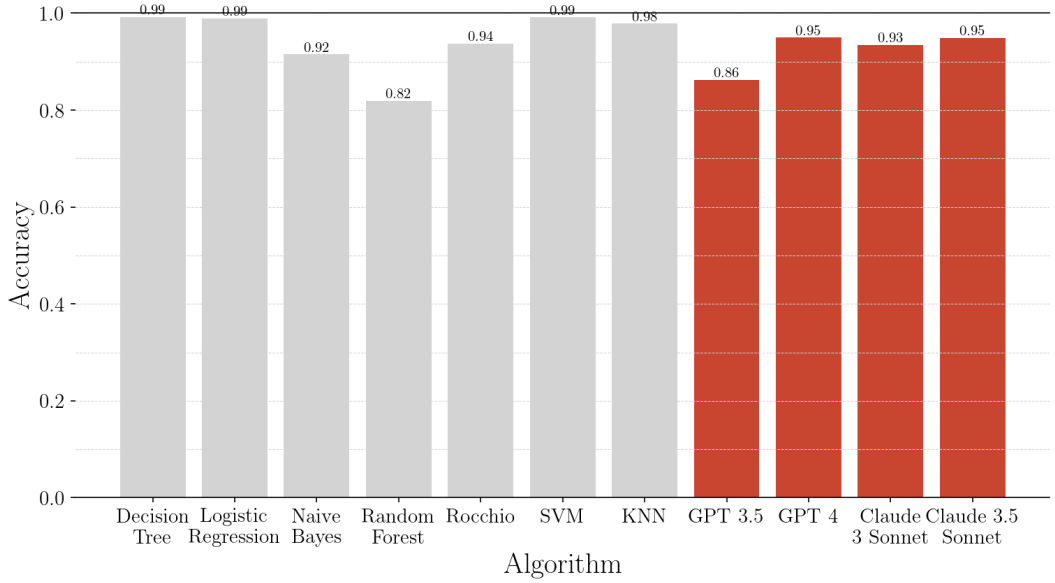
Figure 1: Algorithm's Accuracy in the Test-Train-Split Dataset: Chilean Customs.



(a) HS-6 digit level



(b) HS-4 digit level



(c) HS-2 digit level

Source: Authors' calculations based on Chilean customs data.

Other Performance Metrics. Next, we explore the performance of the various algorithms using other metrics. In multiclass classification problems, we can define a series of alternative measures by class: precision, recall and F1-score. Precision in a given class refers to

the number of instances correctly attributed to such class over the total number of instances where such class was attributed. In other words, low precision indicates a high number of false positives in a given class. Recall refers to the number of instances a given class was correctly attributed divided by the number of times a specific class should have been attributed. Low recall thus indicates a high number of false negatives. Finally, the F1 score is the harmonic mean of precision and recall. It is thus used to provide a measure of the balance between false positives and false negatives. To evaluate the overall performance of the multiclass classification algorithms we need to average the performance across classes. We report both the simple average (macro average) and the average weighted by the number of actual instances of a given class (weighted average). [25]

Table 2 presents a comprehensive analysis of the performance of various classification algorithms on the Chilean test dataset, covering three levels of HS granularity: HS6, HS4 and HS2. At the HS6 level, the Decision Tree and SVM models show the best overall performance, with weighted-average precision, recall and F1-Score all above 0.95.

LLMs such as GPT 4 and Claude 3.5 Sonnet demonstrate remarkable performance, with accuracies of 0.75 and 0.81, respectively. However, their F1-Scores are lower (0.73 and 0.79), indicating a certain imbalance between precision and recall. Despite these strong results, it's worth noting that these performance metrics are slightly lower than those achieved by the best traditional ML algorithms on this train-test dataset. Interestingly, LLMs tend to exhibit significantly better performance in weighted-average metrics than in macro-average metrics, thus indicating that they perform better in relatively common classes.

In more aggregate product categories (at HS4 and HS2), a general improvement in the performance of all models is observed. At the HS4 level, GPT 4 and Claude 3.5 Sonnet improve significantly, reaching accuracies of 0.88 and 0.92 respectively, with corresponding F1-Scores of 0.88 and 0.92. This further suggests an improvement in the precision-recall trade-off for these models. At the HS2 level, GPT 4 and Claude 3.5 Sonnet achieve an accuracy of 0.95, matching or exceeding most traditional algorithms. Their F1-Scores also improve to 0.95,

[25] See Online Appendix A6 for a more detailed explanation of the performance metrics.

indicating a very balanced performance. It is notable that LLMs show substantial improvements in all metrics as granularity decreases. For example, GPT 4 increases its accuracy from 0.75 in HS6 to 0.95 in HS2, and its F1-Score from 0.73 to 0.95.

Table 2: Classification Report for the Chilean Test-Set at Different HS Levels

HS Level	Metric	Dec. Tree	Log. Reg.	Naive Bayes	Rand. Forest	Rocchio	SVM	KNN	GPT 3.5	GPT 4	Claude 3	Claude 3.5
	Accuracy	0.97	0.95	0.79	0.67	0.81	0.95	0.92	0.61	0.75	0.68	0.81
	Macro Avg											
	Precision	0.92	0.91	0.96	0.97	0.69	0.93	0.83	0.54	0.61	0.58	0.64
HS6	Recall	0.88	0.82	0.17	0.09	0.76	0.83	0.65	0.40	0.51	0.47	0.60
	F1-Score	0.88	0.83	0.20	0.10	0.68	0.84	0.66	0.13	0.29	0.20	0.38
	Weighted Avg											
	Precision	0.97	0.95	0.85	0.78	0.86	0.95	0.92	0.79	0.83	0.84	0.86
	Recall	0.97	0.95	0.79	0.67	0.81	0.95	0.92	0.61	0.75	0.67	0.81
	F1-Score	0.97	0.95	0.75	0.59	0.82	0.95	0.92	0.59	0.73	0.66	0.79
	Accuracy	0.99	0.98	0.86	0.74	0.90	0.99	0.96	0.77	0.88	0.85	0.92
	Macro Avg											
	Precision	0.96	0.96	0.96	0.96	0.81	0.97	0.91	0.54	0.67	0.59	0.73
HS4	Recall	0.96	0.94	0.36	0.21	0.87	0.95	0.85	0.70	0.80	0.73	0.83
	F1-Score	0.96	0.95	0.41	0.24	0.83	0.96	0.86	0.42	0.60	0.47	0.66
	Weighted Avg											
	Precision	0.99	0.98	0.91	0.81	0.92	0.99	0.96	0.87	0.91	0.89	0.93
	Recall	0.99	0.98	0.86	0.74	0.90	0.99	0.96	0.77	0.88	0.85	0.92
	F1-Score	0.99	0.98	0.85	0.69	0.91	0.99	0.96	0.77	0.88	0.85	0.92
	Accuracy	0.99	0.99	0.92	0.82	0.94	0.99	0.98	0.86	0.95	0.94	0.95
	Macro Avg											
	Precision	0.98	0.97	0.92	0.90	0.87	0.98	0.94	0.43	0.57	0.51	0.57
HS2	Recall	0.98	0.97	0.71	0.49	0.90	0.98	0.95	0.90	0.91	0.90	0.91
	F1-Score	0.98	0.97	0.77	0.57	0.88	0.98	0.95	0.41	0.55	0.49	0.55
	Weighted Avg											
	Precision	0.99	0.99	0.94	0.85	0.94	0.99	0.98	0.92	0.96	0.95	0.96
	Recall	0.99	0.99	0.92	0.82	0.94	0.99	0.98	0.86	0.95	0.94	0.95
	F1-Score	0.99	0.99	0.91	0.80	0.94	0.99	0.98	0.87	0.95	0.94	0.95

Source: Authors' calculations based on Chilean customs data.

Overall, we find that the LLMs, particularly GPT 4 and Claude 3.5 Sonnet, are very accurate in predicting broad product categories, often outperforming traditional ML algorithms (such as Random Forest and Naives Bayes), even within the test-train-split dataset.

4.2 Results on the External Dataset 1: Trade Transactions from the Paraguayan Customs

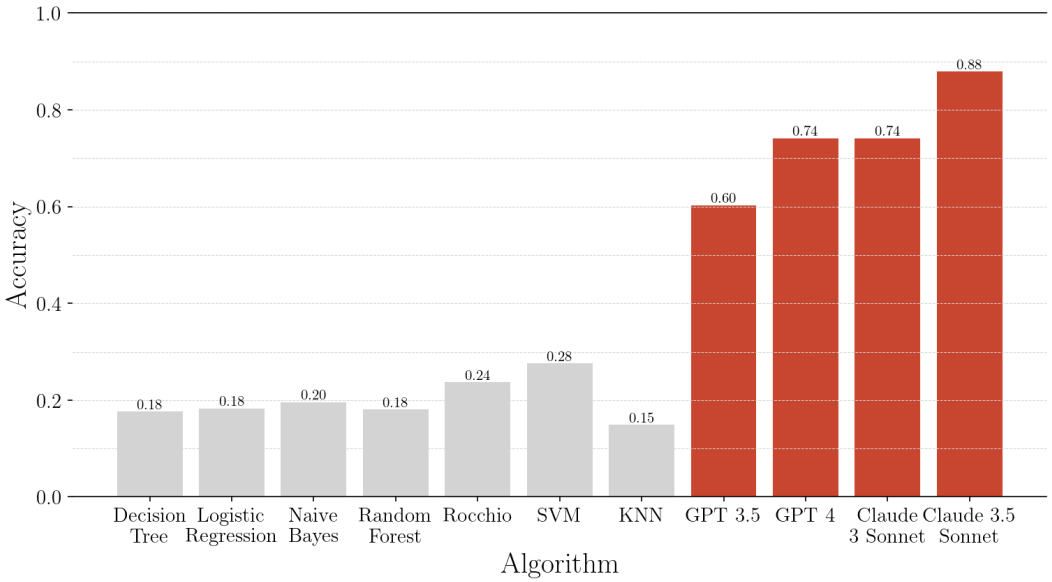
In this subsection we test the ML algorithms and the LLMs outside the dataset on which the ML algorithms were trained. This is key, since the usefulness of such algorithms in real-world applications depends on their external validity. Real data imposes a clear challenge in this regard. It features a variety of product descriptions, including different formats. Hence, a model performing well on the training dataset may not be indicative of how well it will accomplish its classification task in other settings. To explore this, we benchmark the product

classification models using data that was not part of the training dataset. Specifically, we selected a random sample of 10,000 product descriptions from Paraguayan customs records. This allows for a fairer comparison of ML models and LLMs since it confronts all models with text data formats on which none was explicitly trained.

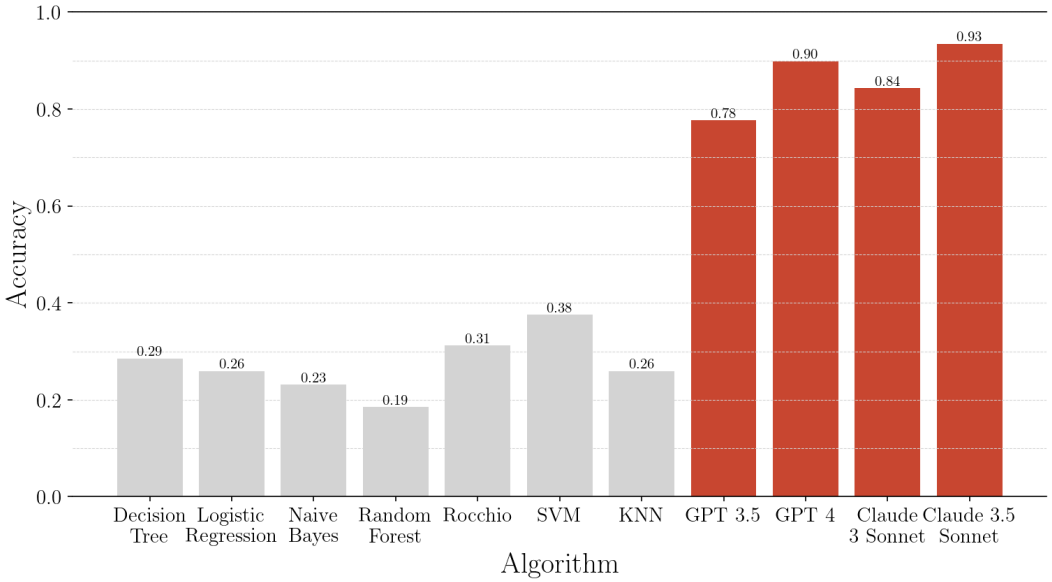
Model Accuracy. Figure 2 shows the accuracy rate attained by the models at different aggregation levels. At the HS6 level, there is a notable contrast between traditional ML models and LLMs. Traditional models such as Decision Tree, Logistic Regression and SVM perform poorly, with accuracy rates ranging between 0.15 and 0.28. In contrast, LLMs exhibit a substantially stronger performance. Claude 3.5 Sonnet stands out with an accuracy of 0.88, followed by GPT 4 and Claude 3 Sonnet, both with 0.74.

As the granularity decreases to HS4, there is a general improvement in the performance of all models, but the gap between LLMs and traditional models remains unaffected. Claude 3.5 Sonnet achieves an accuracy of 0.93, closely followed by GPT 4 with 0.90. The traditional models improve, but still lag far behind, with SVM achieving the best accuracy among them at 0.38. At the HS2 level, the difference in performance narrows, but LLMs still significantly outperform the traditional models. Claude 3.5 Sonnet maintains its lead with an accuracy of 0.94, while Claude 3 Sonnet and GPT 4 achieve 0.93 and 0.92 respectively. It is interesting to note that the Decision Tree shows a significant improvement at this level, reaching an accuracy of 0.73.

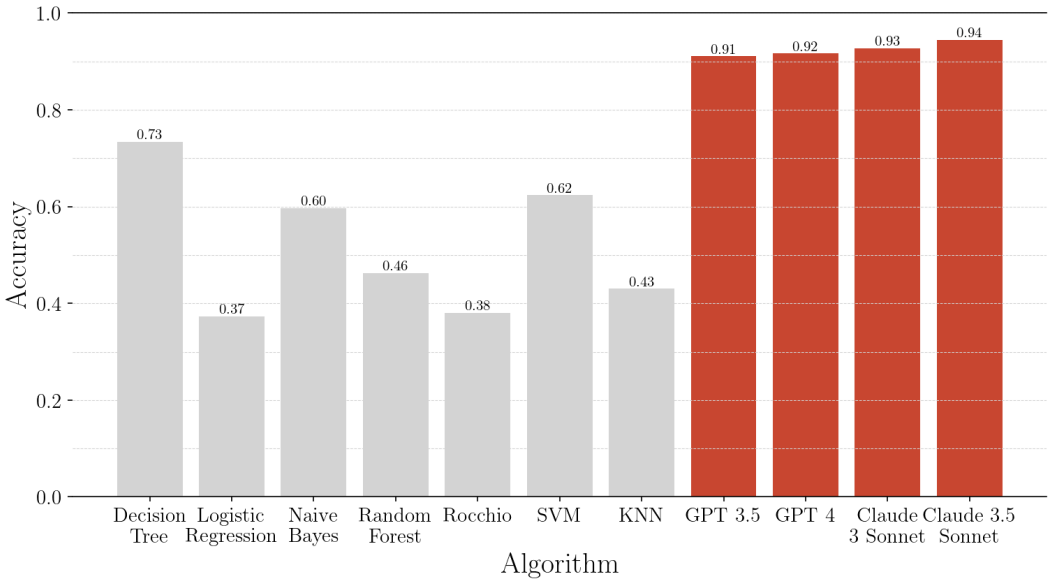
Figure 2: Algorithm's Accuracy in the First External Dataset: Paraguayan Customs.



(a) HS-6 digit level



(b) HS-4 digit level



(c) HS-2 digit level

Source: Authors' calculations based on Paraguayan customs data.

Other Performance Metrics. Table 3 presents a comprehensive analysis of the performance of various classification algorithms on the Paraguayan test data set, covering the three levels of granularity of the Harmonised System (HS): HS6, HS4 and HS2.

Overall, traditional ML models perform poorly in terms of recall and F1-scores. The F1-scores range from 0.11 to 0.28 at the HS 6 digits nomenclature, 0.13-0.37 at HS 4 digits, and 0.41-0.72 at HS 2 digits. Some algorithms achieve relatively high precision but at the cost of very low recall (e.g., Random forest). Macro-averaged performance scores are generally lower than weighted averages, indicating that the models struggle with classes that are relatively infrequent in the dataset.

LLMs show a much better performance. In terms of weighted F1-Score at the HS6 level, Claude 3.5 Sonnet achieves the highest value of 0.87, followed by GPT 4 with 0.76 and Claude 3 Sonnet with 0.73. This indicates that these models are not only accurate but also maintain a good balance between precision and recall. As with ML algorithms, weighted average scores are significantly higher than macro-averaged scores.

Table 3: Classification Report for the Paraguayan Test-Set at Different HS Levels

HS Level	Metric	Dec. Tree	Log. Reg.	Naive Bayes	Rand. Forest	Rocchio	SVM	KNN	GPT 3.5	GPT 4	Claude 3	Claude 3.5
	Accuracy	0.18	0.18	0.20	0.18	0.24	0.28	0.15	0.60	0.74	0.74	0.88
	Macro Avg											
	Precision	0.40	0.35	0.75	0.91	0.45	0.39	0.37	0.42	0.50	0.46	0.56
HS6	Recall	0.56	0.57	0.25	0.11	0.55	0.57	0.54	0.62	0.68	0.69	0.75
	F1-Score	0.10	0.10	0.06	0.04	0.14	0.11	0.08	0.14	0.28	0.23	0.38
	Weighted Avg											
	Precision	0.68	0.64	0.60	0.89	0.72	0.72	0.71	0.83	0.92	0.90	0.95
	Recall	0.18	0.18	0.20	0.18	0.24	0.28	0.15	0.60	0.74	0.74	0.88
	F1-Score	0.20	0.22	0.12	0.11	0.28	0.26	0.18	0.59	0.76	0.73	0.87
	Accuracy	0.29	0.26	0.23	0.19	0.31	0.38	0.26	0.78	0.90	0.83	0.93
	Macro Avg											
	Precision	0.40	0.35	0.73	0.87	0.44	0.39	0.40	0.51	0.60	0.52	0.64
HS4	Recall	0.55	0.58	0.27	0.15	0.55	0.60	0.52	0.79	0.84	0.80	0.87
	F1-Score	0.19	0.20	0.11	0.05	0.22	0.21	0.17	0.39	0.52	0.43	0.56
	Weighted Avg											
	Precision	0.66	0.58	0.57	0.89	0.60	0.72	0.67	0.85	0.94	0.94	0.97
	Recall	0.29	0.26	0.23	0.19	0.31	0.38	0.26	0.78	0.90	0.84	0.93
	F1-Score	0.33	0.28	0.17	0.13	0.33	0.37	0.28	0.77	0.89	0.84	0.93
	Accuracy	0.73	0.37	0.60	0.46	0.38	0.62	0.43	0.91	0.92	0.93	0.94
	Macro Avg											
	Precision	0.46	0.32	0.56	0.70	0.52	0.38	0.33	0.46	0.55	0.49	0.57
HS2	Recall	0.50	0.43	0.27	0.16	0.48	0.53	0.45	0.91	0.90	0.92	0.93
	F1-Score	0.40	0.30	0.25	0.15	0.39	0.39	0.31	0.45	0.52	0.47	0.54
	Weighted Avg											
	Precision	0.81	0.66	0.81	0.89	0.80	0.78	0.68	0.94	0.95	0.95	0.97
	Recall	0.73	0.37	0.60	0.46	0.38	0.62	0.43	0.91	0.92	0.93	0.94
	F1-Score	0.72	0.41	0.59	0.52	0.44	0.67	0.47	0.90	0.91	0.93	0.94

Source: Authors' calculations based on Paraguayan customs data.

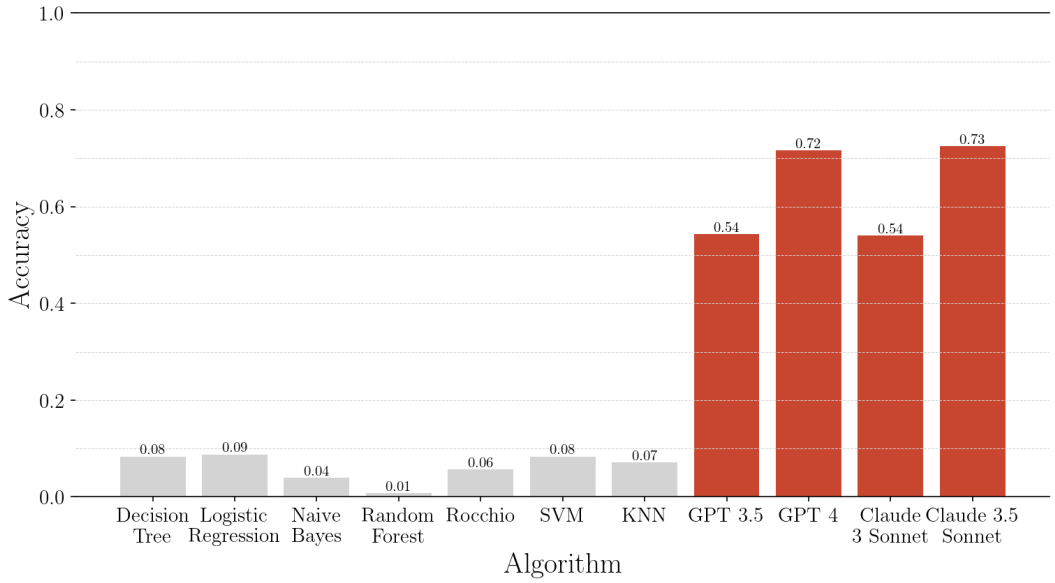
4.3 Results on the External Dataset 2: USDA Organic Product Descriptions

Finally, we assessed the ability of conventional ML algorithms and LLMs to accurately predict HS codes with text formats that differ from the long and highly specific descriptions traditionally used in customs. To do this, we use a set of descriptions of products for which Latin American firms are certified as organic producers according to the United States Department of Agriculture (USDA). These unstructured product descriptions are provided by diverse firms and certification agencies, have a large diversity of formats, and vary significantly in terms of depth and specificity (see Online Appendix A1 for a sample of random descriptions of the three datasets used). This makes them potentially harder to categorize than the average customs product description. Furthermore, this data source does not specify the underlying HS code. Consequently, we do not have a large pool of categorized data that can be used to train tailored ML algorithms. Similar cases can be found in many other data sources, such as cross-border e-commerce shipments, bank transactions, historical trade

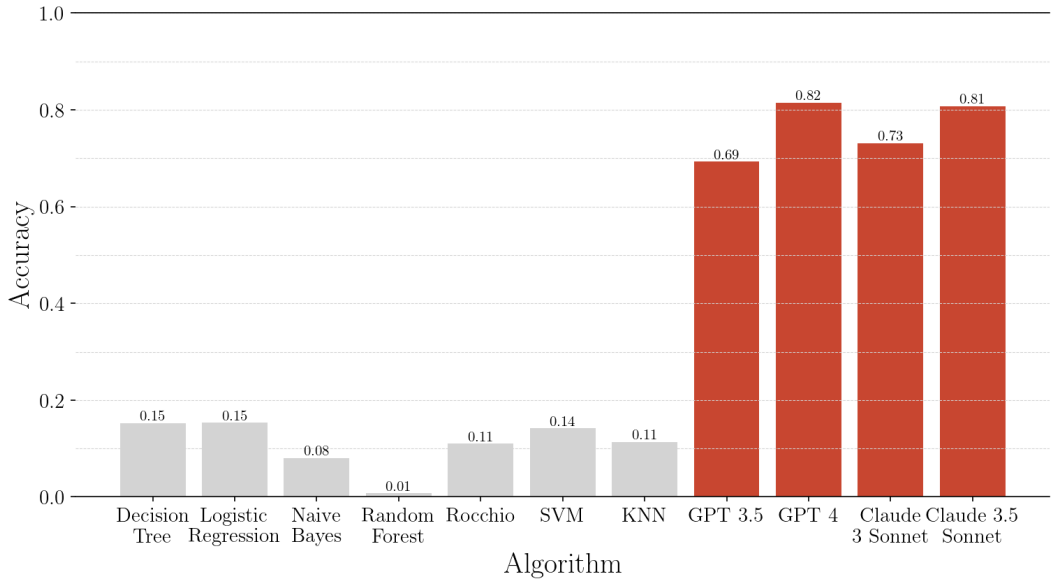
data and survey-based descriptions. For the purpose of this exercise we have drawn a random sample of 1,000 descriptions of USDA certified organic products and classified these by hand into 6-digit HS tariff lines.

Model Accuracy. Figure 3 shows the accuracy rates at the different HS aggregation levels. At HS-6 Claude 3.5 Sonnet achieved the highest accuracy rate -73%-, closely followed by GPT 4 at 72%. GPT 3.5 and Claude 3 Sonnet both achieved 54% accuracy. In contrast, traditional ML models achieved a maximum accuracy rate of 15% (Rocchio model). The differences persist when attempting to classify products at the more aggregate HS 4-digits product nomenclature. In this case GPT 4 led with 82% accuracy, followed by Claude 3.5 Sonnet at 81%, Claude 3 Sonnet at 73%, and GPT 3.5 at 69%. Among the traditional ML algorithms, the maximum accuracy rate was still very low at 26%. At the broadest HS2 level, the performance of traditional methods improved only by 3 to 8 percentage points, with the Decision Tree algorithm reaching 23% accuracy. LLMs maintained their superior performance: GPT 4 achieved the highest accuracy of 92%, followed by Claude 3.5 Sonnet at 89%, Claude 3 Sonnet at 86%, and GPT 3.5 at 81%. Overall, we find that the accuracy differences between ML algorithms and LLMs are even starker in the context of these highly unstructured product descriptions.

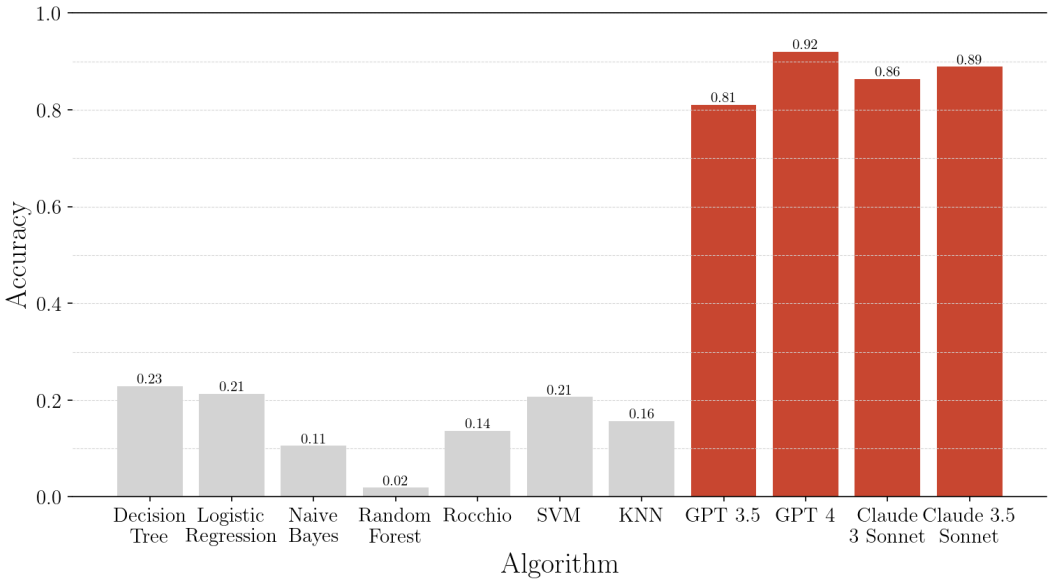
Figure 3: Algorithm's Accuracy in the Second External Dataset: USDA Organic Classification.



(a) HS-6 level



(b) HS-4 level



(c) HS-2 level

Source: Authors' calculations based on USDA data.

Other Performance Metrics. Table 4 presents the full set of performance metrics for the classification problem of the USDA Organic text descriptions at HS6, HS4, and HS2 levels. Overall, traditional ML algorithms perform significantly worse than LLMs. While their precision is relatively high, the recall and accuracy rates are extremely low (weighted averages range between 0.01 and 0.10); thereby indicating a very high share of false negatives. As a result, their F1-scores are very low (0.01-0.09), indicating overall poor performance. Even in the most aggregate categories (HS-2 Chapters), the performance remains weak: the highest performing algorithm (Decision Tree) only attains an F1 score of 0.23.

Claude 3.5 Sonnet has weighted-average precision of 0.84, recall of 0.73 and an F1-score of 0.72. GPT 4 achieved a very similar performance. At the most aggregate category, the performance increases even further, with GPT 4 achieving an F1-score of 0.92. These results underscore the potential of LLMs in classifying product text descriptions across all HS levels, particularly excelling in more detailed classifications where traditional ML algorithms struggle. The ability of LLMs to handle the complexity and variability of the product de-

scriptions from the USDA dataset demonstrates their potential for applications in various domains with highly unstructured or non-standard text data.

Table 4: Classification Report for the USDA Organic Dataset at Different HS Levels

HS Level	Metric	Dec. Tree	Log. Reg.	Naive Bayes	Rand. Forest	Rocchio	SVM	KNN	GPT 3.5	GPT 4	Claude 3	Claude 3.5
	Accuracy	0.09	0.10	0.05	0.02	0.07	0.09	0.08	0.54	0.72	0.54	0.73
	Macro Avg											
	Precision	0.66	0.57	0.84	0.98	0.79	0.65	0.62	0.64	0.69	0.59	0.73
HS6	Recall	0.22	0.31	0.10	0.01	0.12	0.25	0.27	0.58	0.66	0.63	0.73
	F1-Score	0.04	0.03	0.02	0.00	0.02	0.03	0.03	0.33	0.45	0.33	0.55
	Weighted Avg											
	Precision	0.70	0.68	0.83	0.92	0.76	0.70	0.70	0.80	0.84	0.77	0.84
	Recall	0.09	0.10	0.05	0.02	0.07	0.09	0.08	0.54	0.72	0.54	0.73
	F1-Score	0.08	0.09	0.03	0.01	0.07	0.08	0.07	0.53	0.71	0.54	0.72
	Accuracy	0.15	0.15	0.08	0.01	0.11	0.14	0.11	0.69	0.82	0.73	0.81
	Macro Avg											
	Precision	0.60	0.56	0.81	0.95	0.75	0.64	0.61	0.76	0.79	0.69	0.79
HS4	Recall	0.22	0.26	0.10	0.02	0.10	0.20	0.22	0.70	0.79	0.73	0.79
	F1-Score	0.08	0.07	0.04	0.00	0.05	0.07	0.06	0.59	0.69	0.56	0.69
	Weighted Avg											
	Precision	0.72	0.69	0.86	0.96	0.77	0.70	0.67	0.82	0.86	0.79	0.87
	Recall	0.15	0.15	0.08	0.01	0.11	0.14	0.11	0.69	0.82	0.73	0.81
	F1-Score	0.15	0.14	0.07	0.00	0.13	0.13	0.10	0.70	0.81	0.73	0.81
	Accuracy	0.23	0.21	0.11	0.02	0.14	0.21	0.16	0.81	0.92	0.86	0.89
	Macro Avg											
	Precision	0.64	0.62	0.62	0.82	0.70	0.62	0.60	0.79	0.82	0.77	0.81
HS2	Recall	0.20	0.19	0.14	0.07	0.13	0.19	0.16	0.82	0.88	0.82	0.87
	F1-Score	0.15	0.12	0.07	0.02	0.10	0.13	0.10	0.73	0.80	0.72	0.77
	Weighted Avg											
	Precision	0.72	0.66	0.64	0.85	0.72	0.68	0.62	0.87	0.93	0.87	0.91
	Recall	0.23	0.21	0.11	0.02	0.14	0.21	0.16	0.81	0.92	0.86	0.89
	F1-Score	0.23	0.19	0.10	0.00	0.16	0.19	0.14	0.82	0.92	0.87	0.89

Source: Authors' calculations based on USDA Organic dataset.

4.4 Extensions and Robustness Checks

In this section we carry out a series of extensions and robustness checks of our baseline multiclass classification models.

Hallucination of HS Codes in LLMs. In the context of language models, 'hallucination' refers to the phenomenon where the model generates information that is fictitious or not supported by the training data. In our specific case, hallucination occurs when the model generates HS codes that do not exist in the official nomenclature. In this section, we thus analyse how frequently LLMs attribute non-existent HS codes. For this purpose, we identify whether the HS6 codes generated by the LLMs for the classification of the Chilean data coincide with existing HS codes according to the official nomenclature. We report two metrics: (i) the number of unique hallucinated codes and (ii) the share of instances where the LLM

predicts a non-existent code (hallucination rate).

Table 5: Hallucination in Product Classification in LLMs — HS6 codes, Chilean data

Model	Unique Hallucinated HS-6 Codes	Hallucination Rate
GPT 3.5	51	1.93%
GPT 4	14	1.70%
Claude-3 Sonnet	33	3.25%
Claude-3.5 Sonnet	14	0.31%

Source: Authors' calculations based on Chilean customs data.

Our results, presented in Table 5, indicate notable differences across LLMs. GPT 4 and Claude 3.5 Sonnet both produce 14 unique hallucinated codes, while Claude Sonnet 3 produces 33 and GPT 3.5 a total of 49. Claude 3.5 has the lowest hallucination rate (0.31%), followed by GPT 4 (1.70%), GPT 3.5 (1.93%) and, finally, Claude 3 (3.25%). Newer versions of the models thus seem to have significantly reduced the hallucination rates. The case of Claude is particularly remarkable, with a large improvement between the 3 and the 3.5 versions.

Consistent with the accuracy rates shown before, hallucination rates at HS4 shows much smaller numbers. GPT 3.5 has the highest hallucination rate (0.11%), but with only 8 codes, followed by Claude 3 Sonnet with only one hallucinated code and 0.01% of hallucination rate. Both advanced models (GPT 4 and Claude 3.5) show 0.00% hallucination rate, which is a big improvement in their performance.

Table 6: Hallucination in Product Classification in LLMs — HS4 codes, Chilean data

Model	Unique Hallucinated HS-4 Codes	Hallucination Rate
GPT 3.5	8	0.11%
GPT 4	0	0.00%
Claude-3 Sonnet	1	0.01%
Claude-3.5 Sonnet	0	0.00%

Source: Authors' calculations based on Chilean customs data.

Performance Metrics by Class. Next, we explore the performance of the LLMs across different product categories. Online Appendix A7 presents the differences in terms of accu-

racy across HS-2 Chapters. The performance is highly correlated across LLMs: for instance most models struggle with HS Chapters 5 ("Other products of animal origin, not elsewhere specified") and 14 ("Vegetable plaiting, vegetable products not included elsewhere"). These two chapters include categories with products "not classified elsewhere", which may complicate automatic classification due to the lack of clear boundaries and the ambiguity of such category. On the other hand, most models perform very well in HS Chapters 2 ("Meat products"), 3 ("Fish and crustaceans"), 10 ("Cereals") and 22 ("Beverages and spirits"), which are much more distinct and clearly delimited. Online Appendix A8 shows other performance metrics by HS-2 Chapter including recall, precision and F1-scores.

Finally, Online Appendix A9 shows the accuracy distribution in more granular categories. Due to the large number of categories, we represent the distribution through accuracy percentiles (e.g. P50 represents the median accuracy rate across classes). Overall, we see that high-performing models such as Claude 3.5 Sonnet and GPT 4 tend to especially outperform other models in low percentiles of the distribution. Whereas in most models, the P90 tends to have accuracy rates above 95%, the largest differences happen in the low-accuracy classes: the 10th percentile of the distribution (P10) in Claude 3.5 corresponds to an accuracy rate of 50%, whereas for Claude 3.0 it corresponds to just 5%. Thus, new LLMs seem to improve their overall performance by increasing accuracy rates in classes where prior versions were under-performing. In addition to this, a high level of accuracy continues to be observed within the training data for the ML models, especially for Decision Tree and SVM.

Training Sample Balanced by HS2 Chapter. In the exercises in Sections 4.1-4.3 we trained ML models using a random sample of 1M Chilean observations across all observations in Chapters 1-22. This approach, while representative of the actual distribution of Chilean trade, leads to a highly unbalanced sample: HS Chapters 08 ("Edible fruits and nuts") and 22 ("Beverages and spirits"), which are products frequently exported by Chile, jointly cover more than 55% of the observations in our sample (see Table A10a). In Online Appendix A10 we show the results from estimating the ML models with in a sample where we try to balance

as much as possible across HS 2-digits Chapters. To construct a balanced sample of 1 million observations, we randomly draw ∼45,000 observations per Chapter.[26] In some cases, the original data has less than 45,000 observations, and thus we draw all available observations.

Overall, balancing across HS Chapters does not significantly improve the performance of the ML algorithm in the train-test-split sample (Chilean customs). For the first external dataset (Paraguayan customs) there is a relatively small improvement in macro-averaged recall (of approximately 5 percentage points), but the overall accuracy rate is slightly lower. For the second external dataset (USDA organic) there is a marginal increase in accuracy (of approximately 3 percentage points). Thus, balancing across Chapters leads to, if anything, very small gains in performance. LLM models continue to outperform by a very large margin traditional ML algorithms outside the test-train database.

Swapping the Train-Test-Split Database: Training with Paraguayan Data. Next, we explore the robustness of our results by switching the train-test-split dataset, and training the ML models with the Paraguayan customs data. We follow the same data preparation procedure as in Section 3.1 and estimate the models described in Section 3.2; thus mirroring the procedure used in our baseline that uses Chilean data to train the ML algorithms. The results can be found in Online Appendix A11.

Our findings are very similar to those in the baseline model. The traditional ML algorithms perform very well in the alternative test-train-split dataset (Paraguayan customs), with accuracy rates around 95% in Decision Tree, Logistic Regression and SVM. The performance, however, drops dramatically when we use the models trained on Paraguayan data on our Chilean customs product descriptions. In this setting the best performing ML model (SVM with 49% accuracy rates and a weighted-average F1-Score of 0.47) lags significantly behind LLM models. For instance, Claude 3.5 Sonnet achieves 81% accuracy and 0.79 weighted-average F1-Score. In the USDA organic dataset -which, as explained above, has highly unstructured text data with relatively coarse product descriptions-, the difference

[26]Note that 1 million observations divided by 22 chapters leads to 45,454 observations per Chapter

widens even further. Our results thus seem to be robust to the specific choice of train-test-split database.

Nested Prompts. Our baseline LLM prompt asks the models to directly provide an estimate of the HS-6 digits product code. Alternatively, we explore a "nested" version of the prompt. In this case, we first prompt the LLM to assign a 2-digit HS code based on the product description. Subsequently, we ask it to provide a 4-digit HS code among those that belong to the aggregate 2-digit HS code category that it previously identified. Finally, we requested the model to provide a 6-digit HS code belonging to the prior 4-digit HS code that it had identified.

The specific details for this LLM nested prompt can be found in Online Appendix A12, along with the results for this alternative prompting strategy. Interestingly, we find that nested prompting decreases the accuracy of the classification. One potential explanation is that this nested structure constrains the models, thus leading to more frequent hallucinations of non-existent HS codes. The same exercise conducted for Table 5 showed that the Nested prompts have a higher number of hallucinated HS-6 codes (77) and slightly higher hallucination rates (2.10%).

Other LLMs and LLM Sub-Sample Accuracy. Finally, we benchmark the performance of other LLMs. For this purpose we use Poe, an open platform to explore LLMs. We use a smaller random sample of 100 product descriptions, which we obtain from Chilean customs. We test the accuracy rates of five additional LLMs: Bard, Claude-Instant, LLaMa 2, Solar and PaLM 2. Claude Sonnet 3.5 and GPT 4 significantly outperform these alternative LLMs, the highest performing of which is Bard (see Online Appendix A13). [27]

[27] Relatedly, in Online Appendix A14 we explore how stable are the estimated accuracy rates for LLMs when we increase the sample size by one order of magnitude. For this purpose, we first obtain a random sample of 1000 product descriptions and randomly divide it into 10 groups of 100 descriptions. We find very similar estimated accuracy rates across the 10 groups, with a very low standard deviation among them.

5 Discussion and Conclusions

LLMs demonstrate high accuracy rates when classifying products according to the HS nomenclature. While traditional ML algorithms performed well on their training dataset (even some of them outperformed LLMs), their performance sharply declined when tested on external data. In such external validity tests, LLMs significantly outperformed these algorithms. Claude 3.5 Sonnet consistently achieved the highest accuracy across different datasets and HS levels, closely followed by GPT 4. The superior performance of LLMs was evident despite the fact that our ML models were trained with 1 million observations of high-quality customs product descriptions and then tested on high-quality descriptions from another customs agency from a country. This highlights the robustness and generalizability of LLMs in product classification tasks.

Another major advantage of LLMs is their ability to work with product descriptions in different languages. Throughout our analysis, we used data in English and Spanish, but LLMs are likely to perform very well across many other languages in which large amounts of data are publicly available (e.g., Chinese, French, German, etc.). Importantly, they are also able to handle regional variants of the same language successfully. One interesting example from our study was *Physalis peruviana*, a fruit typically known as “goldenberry” in English. Our Chilean training data refers to them as “uchuva,” but the fruit goes by other names in different countries: "camambu" in Paraguay, “uvilla” in Ecuador, “aguaymanto” in Peru, and “fisalis” in Spain. ML algorithms trained on the Chilean data failed to identify these regional variations and thus misclassified the product, whereas GPT and Claude models, trained on a much wider set of texts, recognized the fruit and classified it properly. This is an example of how the wide training dataset of LLMs allows them to outperform standard ML algorithms. LLMs can thus be very useful in comprehensive unilateral, regional, and multilateral trade policy initiatives involving product classifications over time and across countries.

LLMs are also significantly simpler to use and implement since they do not require data-cleaning and preprocessing routines. Performing these tasks with traditional ML algorithms

can be rather time-consuming and resource-intensive, especially those related to feature extraction.[28] While an API is necessary to work at scale with LLMs, the standard interface enables the classification functionality to be integrated easily into existing systems or applications. In our analysis, we worked with the base models, without making further adjustments, but LLMs could also be adapted for use with specific data. Fine-tuning LLMs with trade data may help them be more effective at product classification at scale.

In terms of cost, the models we used (GPT 3.5, GPT 4, and Claude models) are relatively inexpensive, except for very large tasks.[29] Importantly, open-source LLMs are also becoming increasingly competitive and can be expected to perform very well in large-scale product classification tasks in the short term. Benchmarking automatic product classification at a larger scale across a wide range of LLMs, including both commercial and open-source models, and exploring various fine-tuning methods, therefore, remains an important avenue for future research. This could help identify the most cost-effective and accurate solutions for different scales and types of classification tasks.

[28] We also assessed the model against a manual classification carried out by a research assistant (RA) using a sample of 100 observations. The results indicate that, while the RA's accuracy was slightly above that of GPT 3.5 at the HS 6-digit level, the difference fades when more aggregate classification levels are considered. At the 2-digit HS level, GPT 3.5 performed slightly better than the RA. It is worth stressing that while the RA needed four hours to accomplish the task, GPT 3.5 completed it in just one minute. This suggests that there is potentially a tradeoff between accuracy and time for highly disaggregated classifications in small samples. The terms of this tradeoff are highly likely to change as the number of observations increases, with GPT 3.5 clearly emerging as the better approach for large samples, especially given that human working time increases at a nonlinear rate due to marginal decreasing returns.

[29] In our work, we used OpenAI's GPT 3.5 ("gpt-3.5-turbo") and GPT 4 ("gpt-4") and Anthropic's Claude 3 Sonnet and Claude 3.5 Sonnet. Without going into the billing system works in detail, our estimate is that the total cost of classifying a dataset of 10,000 standard customs product descriptions is approximately $2.5 for GPT 3.5 and $40 for GPT 4 while for the Anthropic's models it is $4 each. (see the OpenAI's pricing and Anthropic's pricing).

References

Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. *Mining Text Data*, 163-222.

Aly, M. (2005). Survey on multiclass classification methods. *Neural Networks, 19*, 1-9.

Biswas, S. S. (2023). Role of ChatGPT in Public Health. *Annals of Biomedical Engineering, 51*(5), 868-869.

Chen, H., Van Rijnsoever, B., Molenhuis, M., van Dijk, D., Tan, Y. H., & Rukanova, B. (2021). The use of machine learning to identify the correctness of HS Code for customs import declarations. *2021 Institute of Electrical and Electronics Engineers 8th International Conference on Data Science and Advanced Analytics*, 1-8.

Fujiyoshi, H., Hirakawa, T., & Yamashita, T. (2019). Deep learning-based image recognition for autonomous driving. *International Association of Traffic and Safety Sciences, 43*(4), 244-252.

Hansen, A. L., & Kazinnik, S. (2023). Can Chatgpt decipher Fedspeak? *Social Science Research Network*.

Hansen, S., Lambert, P. J., Bloom, N., Davis, S. J., Sadun, R., & Taska, B. (2023). Remote work across jobs, companies, and space. *National Bureau of Economic Research* (Working Paper No. 31007).

He, M., Wang, X., Zou, C., Dai, B., & Jin, L. (2021). A commodity classification framework based on machine learning for analysis of trade declarations. *Symmetry, 13*(6), 964.

Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., & Kasneci, G. (2023). Chatgpt for good? on opportunities and challenges of Large Language Models for education. *Learning and Individual Differences, 103*.

Ko, H., & Lee, J. (2023). Can Chatgpt improve investment decisions? A portfolio management perspective. *Social Science Research Network*

Kocon, J., Cichecki, I., Kaszyca, O., Kochanek, M., Szydło, D., Baran, J., Kazienko, P. (2023). Chatgpt: Jack of all trades, master of none. *Information Fusion, 101861*.

Korinek, A. (2023). Language models and cognitive automation for economic research. *National Bureau of Economic Research* (Technical Report No. 30957).

Korinek, A. (2023). Generative AI for economic research: Use cases and implications for economists. *Journal of Economic Literature, 61(4)*, 1281-1317.

Korinek, A. (2024) LLMs Level Up—Better, Faster, Cheaper: June 2024 Update to Section 3 of 'Generative AI for Economic Research: Use Cases and Implications for Economists,' *Journal of Economic Literature 61(4)*.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information, 10*(4), 150.

Lai, Y. (2019). A comparison of traditional machine learning and deep learning in image recognition. *Journal of Physics: Conference Series*, 1314.

Lee, J. K., Choi, K., & Kim, G. (2021). Development of a natural language processing based deep learning model for automated HS code classification of the imported goods. *Journal of Digital Contents Society, 22*(3), 501-508.

Lopez-Lira, A., & Tang, Y. (2023). Can Chatgpt forecast stock price movements? Return predictability and Large Language Models. *ArXiv Preprint.*

Lund, B. D., & Wang, T. (2023). Chatting about ChatGPT: how may AI and GPT impact academia and libraries? *Library Hi Tech News, 40*(3), 26-29.

Marra de Artiñano, Riottini Depetris, F., I., Scattolo, G., Volpe Martincus, C., & Zavala, L. (2024). The value of organic certifications. *Inter-American Development Bank Working Paper*. (Forthcoming).

Mitra, A. (2020). Sentiment analysis using machine learning approaches (Lexicon based on movie review dataset). *Journal of Ubiquitous Computing and Communication Technologies, 2*(3), 145-152.

National Customs Service of Chile (2023). Base de datos de operaciones de ingreso del Servicio Nacional de Aduanas de Chile. Database available at the following URL: `https://www.aduana.cl/base-de-datos-operaciones-de-ingreso/aduana/2018-12-28/102736.html`

National Customs Agency of Paraguay (2023). Portal de datos abiertos de aduanas de la

Dirección Nacional de Ingresos Tributarios. Database available at the following URL: https://www.aduana.gov.py/?page_id=14523

Noy, S., & Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. *Social Science Research Network*.

Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. *Institute of Electrical and Electronics Engineers Transactions on Neural Networks and Learning Systems, 32*(2), 604-624.

Ruder, D. (2020). Application of Machine Learning for Automated HS-6 Code Assignment. *Master's thesis, University of Tartu, Institute of Computer Science.*

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science, 2*(3), 160.

Spichakova, M., & Haav, H. M. (2020). Application of Machine Learning for Assessment of HS Code Correctness. *Baltic Journal of Modern Computing, 8*(4), 698-718.

Turhan, B., Akar, G. B., Turhan, C., & Yukse, C. (2015). Visual and textual feature fusion for automatic customs tariff classification. *2015 Institute of Electrical and Electronics Engineers International Conference on Information Reuse and Integration*, 76-81.

USDA Integrity Dataset (2023). Organic Integrity Database of the United States Department of Agriculture. Database available at: https://organic.ams.usda.gov/integrity/Default

Xu, C.-J., & Li, X.-F. (2019). Research on the Classification Method of HS Code Products Based on Deep Learning. *Modern computer, 1*, 13-21.

Yang, K., Ji, S., Zhang, T., Xie, Q., & Ananiadou, S. (2023). On the evaluations of chatgpt and emotion-enhanced prompting for mental health analysis. *ArXiv Preprint.*

Yang, K.-C., & Menczer, F. (2023). Large Language Models Can Rate News Outlet Credibility. *ArXiv Preprint.*

Zheng, L., Chiang, W. L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., ... & Stoica, I. (2024). Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems, 36*

Online Appendix

A1 Information Sources

Table A1: Summary of Databases Used and Description of the Fields

Databases Used	**Availability of Data**	**URL of the Dataset**	**Fields Used**	**Textual Description**
Chile	Custom Transactions from Chile: 2009–2021	https://www.aduana.cl/base-de-datos-operaciones-de-ingreso/aduana/2018-12-28/102736.html	NOMBRE ATRIBUTO1 ATRIBUTO2 ATRIBUTO3 ATRIBUTO4 ATRIBUTO5 ATRIBUTO6 CODIGOARANCEL	Product Description 1 Product Description 2 Product Description 3 Product Description 4 Product Description 5 Product Description 6 Product Description 7 HS-SA
Paraguay	Custom Transactions from Paraguay: 2009–2021	https://www.aduana.gov.py/?page_id=14523	MERCADERIA POSICION	Product Description HS-SA
USDA	United Stated Department of Agriculture Integrity	https://organic.ams.usda.gov/integrity/Default	Certified Products Under CROPS Scope	Product Description

A2 Product Description Examples

Table A2a: Sample of 10 Randomly Chosen Organic Product Descriptions

Original Product
Ungurahui (Oenocarpus Bataua)
soy beans
Plátanos/Bananos - 1 Traboar_Finca Genoveva (F)
Banana puree acidulated deep frozen
Organic aseptic concentrate soursop pulp
Organic white corn powder
Banana puree without seeds
Organic coco
Safflawer
Maca flour - pre cooked

Source: Authors' elaboration based on USDA.

Table A2b: Sample of 10 Randomly Chosen Chilean Product Descriptions

Original Product
Rainbow trout hg caleta bay mar spa-f size 2-9 lbs oncorhynchus mykiss frozen iqf gutted premium
Red wines with denomination of origin vina casa silva cabernet sauvignon alcohol content 14.70% volatile acidity 0.50% harvest 2008 720 bottles of 0.75 liters
Fresh traditional blueberries santa olga-f blue kordia size spd-pd
Agrosuper-f pork fat small frozen
Live worms tebex-f tebo for fishing bait
Fresh salmon fillet cookeaqua-f salmo salar size 3-4lbs premium quality, with skin, boneless, with scales, trim d, raw, chilled-refrigerated, box of
Agromarin-f lamb chop 8 ct (iw), bone-in frozen
San clemente-f raspberry concentrate juice 600 gallons (93)
Belgioioso burrata 6/8 oz (2-4 oz cup) is buffalo milk cream cheese.
Fresh cherries lo garces-f regina size j-sj-xl-sjd-jd-xld-p, 5.0 kn carton, in 762 boxes

Source: Authors' elaboration based on Chilean customs data.

Table A2c: Sample of 10 Randomly Chosen Paraguayan Product Descriptions

Original Product
Kilos of dried orange leaves, in 31 bales of approximately 25 kg each
Others in 2,100 wooden boxes with 24 kilos net and 26 kilos gross each containing fresh bananas in their natural state for consumption
1) pallets containing 42) boxes with 644) kilo mozzarella cheese trebol
Others; it is: 1,014,000 kilograms of rice in 50 kg bags each, 2016/2017 harvest
1,200 boxes of frozen beef
150 one hundred and fifty metric tons of Paraguayan soybean in bulk
100,000 kilos of cane sugar
Offal
15 boxes of pepper seasonings
330 mt. (three hundred and thirty metric tons) Paraguayan corn in bulk, destination: Montevideo - Uruguay

Source: Authors' elaboration based on Paraguayan customs data.

A3 Preparation Steps of Descriptions

Table A3: Preparation Steps of a Random Selected Description

Step	Result
Initial description	FROZEN DOUGHS EUROPASTRY-F CODE-81299 BERLIDOTS BOMBOM FOOD PREPARATION BASED ON WHEAT FLOUR AND WATER IN BOXES OF 36 UNITS FOR HUMAN CONSUMPTION
Text preparation	['FROZEN', 'DOUGHS', 'EUROPASTRY-F', 'CODE-81299', 'BERLIDOTS', 'BOMBOM', 'FOOD', 'PREPARATION', 'BASED', 'ON', 'WHEAT', 'FLOUR', 'AND', 'WATER', 'IN', 'BOXES', 'OF', '36', 'UNITS', 'FOR', 'HUMAN', 'CONSUMPTION']
Lowercase	['frozen', 'doughs', 'europastery-f', 'code-81299', 'berlidots', 'bombom', 'food', 'preparation', 'based', 'on', 'wheat', 'flour', 'and', 'water', 'in', 'boxes', 'of', '36', 'units', 'for', 'human', 'consumption']
Removal of non-ASCII characters	['frozen', 'doughs', 'europastery-f', 'code-81299', 'berlidots', 'bombom', 'food', 'preparation', 'based', 'on', 'wheat', 'flour', 'and', 'water', 'in', 'boxes', 'of', '36', 'units', 'for', 'human', 'consumption']
Converting numbers written in words to digits	['frozen', 'doughs', 'europastery-f', 'code-81299', 'berlidots', 'bombom', 'food', 'preparation', 'based', 'on', 'wheat', 'flour', 'and', 'water', 'in', 'boxes', 'of', '36', 'units', 'for', 'human', 'consumption']
Stop-word removal	['frozen', 'doughs', 'europastery-f', 'code-81299', 'berlidots', 'bombom', 'food', 'preparation', 'based', 'wheat', 'flour', 'water', 'boxes', '36', 'units', 'human', 'consumption']
Lemmatization	['frozen', 'dough', 'europastery-f', 'code-81299', 'berlidot', 'bombom', 'food', 'preparation', 'base', 'wheat', 'flour', 'water', 'box', '36', 'unit', 'human', 'consumption']
Removing words that are not in English or Spanish	['frozen', 'dough', 'code', 'berlidot', 'bombom', 'food', 'preparation', 'base', 'wheat', 'flour', 'water', 'box', '36', 'unit', 'human', 'consumption']
English and Spanish noise removal	['frozen', 'dough', 'berlidot', 'bombom', 'food', 'preparation', 'base', 'wheat', 'flour', 'water', 'box', '36', 'unit', 'human', 'consumption']

Source: Authors' calculations based on Chilean customs data.

A4 Randomization and Pre-Randomization Process

In this Appendix, we describe the randomization process used to train the models. As explained in the main text, after cleaning the Chilean customs database, one million observations are randomly selected to train the models.

```
# Read the file
df = pd.read_csv(cleaned_data.csv)

# There are still some NA in some rows. We delete them
df = df.dropna()

# We delete the rows where HS-SA is 0.0 (no code assigned)
df = df.where(df[HS-SA] != 0.0).dropna()

# We delete NA from the information column
df = df.dropna(subset=[information])

df = df.drop_duplicates(subset=[operation, information, HS-SA])
```

After a few additional steps, which refer to the elimination of special codes (such as 160000), we select those with codes between HS 1 and HS 22, inclusive, and generate the simple randomization to obtain the final number of observations through the Python random sample function *sample*.

```

df['HS-SA'] = df['HS-SA'].astype(int)
df = df.loc[df['HS-SA'] < 23000000]

# Remove rows where HS-SA is equal to '160000' as erroneous
df = df.loc[df[HS-SA] != 160000, :]

# Sample of million observations
df = df.sample(1000000)
```

A5 LLMs Prompts

A5.1 GPT Prompt

```
def assign_code_forced(row, column):
    text = row[column]
    modelo = "gpt-3.5-turbo"
    response = openai.ChatCompletion.create(model=modelo,
    messages=[
        {"role": "system", "content": "You are a helpful assistant that assigns
            product codes in the HS6 product nomenclature categorization."},
        {"role": "user", "content": f'Please assign the harmonized system code
            number in the HS6 for the following description:"{texto}". Return "
            Code: number here". If you are unsure of the classification, provide
            your best possible option'}],
    temperature=0.1)
    assigned_code = response['choices'][0]['message']['content']
    return assigned_code
```

A5.2 Claude Prompt

```
def claude_3_5_code_assigner(fila, columna_descripcion):
    texto = fila[columna_descripcion]
    modelo = "claude-3-5-sonnet-20240620"
    response = client.messages.create(
        model=modelo,
        max_tokens=1000,
        temperature=0,
        system="You are a helpful assistant that assign codes in the Harmonized
            System of UN Comtrade and answer only the 'Code: HS6' part.",
        messages=[
            {
                "role": "user",
                "content": [
                    {
                        "type": "text",
                        "text": f'Please assign the harmonized system code number
                            in the HS6 for the following description:"{texto}".
                            Return "Code: number here" only. If you are unsure of
                            the classification, provide your best possible option'
                    }
                ]
```

```
            }
        ]
    )
    codigo_asignado = response.content[0].text
    return codigo_asignado
```

A6 Multiclass Classification Metrics

In this Appendix, we provide a brief explanation of the performance metrics used to evaluate ML and LLM algorithms in multiclass classification problems.

Below we have a confusion matrix, which defines the categories that will be used to formulate the metrics.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

For example, suppose that we have a product description that belong to the class "HS2 Code 12", describing "Oil seeds and oleaginous plants; miscellaneous grains, seeds and fruit; industrial or medicinal plants; straw and fodder". In this context, we can have the following cases:

- A True Positive (TP): The model predicts that a product is an oilseed (HS2 12) and it actually is.
- A False Negative (FN): The model predicts that a product is not an oilseed (HS2 12), but it actually is.
- A False Positive (FP): The model predicts that a product is an oilseed (HS2 12), but in fact it is not.
- A True Negative (TN): The model predicts that a product is not an oilseed (HS2 12) and it really is not.

Multi-class **accuracy** is defined as the proportion of correctly predicted instances to the total number of instances, which can be understood as the average of correct predictions. The formula for multi-class accuracy is given by:

$$\textbf{accuracy} = \frac{1}{N} \sum_{k=1}^{|G|} \sum_{x:g(x)=k} I(g(x) = \hat{g}(x)) \tag{1}$$

Where N denotes the total number of observations, G represents the set of all classes, and I is the indicator function that returns one when the predicted class matches the true class and zero otherwise. This is our preferred metric, as it provides a direct measure of model performance by indicating how many instances are correctly classified.

Precision is a metric specifically focused on positive *predictions*. It quantifies the proportion of positive predictions that are actually correct. In our example, it quantifies the proportion of predictions for Chapter 12 that are actually correct. In other words, precision measures the exactness or quality of the positive predictions made by the model. Mathematically, precision is defined as:

$$P = \frac{TP}{TP + FP} \tag{2}$$

where TP denotes the number of correctly predicted positive instances, and FP represents the number of negative instances that were incorrectly classified as positive. A high precision indicates that the model has a low false positive rate, which is crucial in applications where the cost of false positives is high.

Recall is defined as the proportion of actual positive *instances* that are correctly identified by the model. In our example, it quantifies the proportion of Chapter 12 that is correctly predicted as Chapter 12. Formally recall is thus given by:

$$R = \frac{TP}{TP + FN} \tag{3}$$

where TP is the number of correctly predicted positive instances, and FN represents the number of positive instances that were incorrectly classified as negative. A high recall indicates that the model has a low false negative rate, which is critical in applications where missing a positive instance has significant consequences.

The **F1-score** provides a balance between precision and recall by considering both metrics simultaneously. The F1-score is particularly useful in situations where there is an uneven class distribution, or where false positives and false negatives carry different implications. It is defined as the harmonic mean of precision and recall, and is given by the following formula:

$$F = 2 \times \frac{P \times R}{P + R} \tag{4}$$

where P represents precision and R represents recall. The harmonic mean ensures that the F1-score is high only when both precision and recall are high, providing a more balanced measure of a model's performance than either metric alone.

As presented above, precision, recall and the F1-score refer to the performance of the model in a specific class. To assess the performance of the model as a whole, we need to average across classes. One option is to carry out a simple average, often called **macro-average**. The macro average treats all classes equally, regardless of their size. It calculates the average performance for each class and then averages these values. The macro averaged precision, recall, and F1-score are defined as follows:

$$P_{\text{macro}} = \frac{1}{|G|} \sum_{i=1}^{|G|} \frac{TP_i}{TP_i + FP_i} \tag{5}$$

$$R_{\text{macro}} = \frac{1}{|G|} \sum_{i=1}^{|G|} \frac{TP_i}{TP_i + FN_i} \tag{6}$$

$$F_{\text{macro}} = 2 \times \frac{P_{\text{macro}} \times R_{\text{macro}}}{P_{\text{macro}} + R_{\text{macro}}} \tag{7}$$

where $|G|$ is the total number of classes, TP_i is the number of true positives for class i, FP_i is the number of false positives for class i, and FN_i is the number of false negatives for class i. The macro average is particularly useful when the goal is to achieve good performance across all classes rather than being dominated by the performance of the majority class.

Alternatively, and especially in situations where class distributions are highly imbalanced, a **weighted average** can be used. This metric assigns a weight w_k to each class k such that $w_k = \frac{1}{|G|}$ for all $k \in \{1, \ldots, G\}$. This approach ensures that each class contributes equally to the overall accuracy, irrespective of its proportion in the dataset. The formula for weighted accuracy is given by:

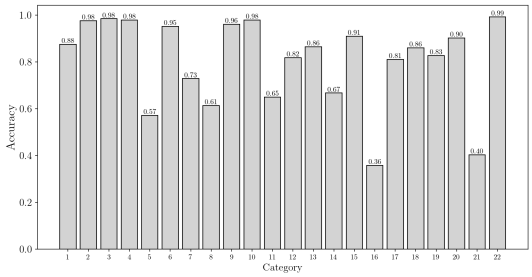
$$\textbf{weighted average} = \frac{1}{N} \sum_{k=1}^{|G|} w_i \sum_{x:g(x)=k} I(g(x) = \hat{g}(x)) \tag{8}$$

In this context, w_i represents the weight assigned to each class i, N is the total number of observations, G is the set of all classes, and I is the indicator function that returns one when the predicted class matches the true class, and zero otherwise.

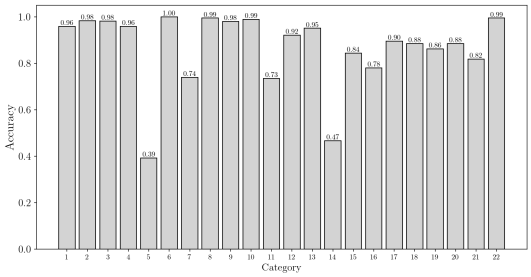
A7 Accuracy in HS-2 Chapters

In this section, we show the accuracy of the LLM algorithms at the HS-6 digits for each HS Chapter 1–22.

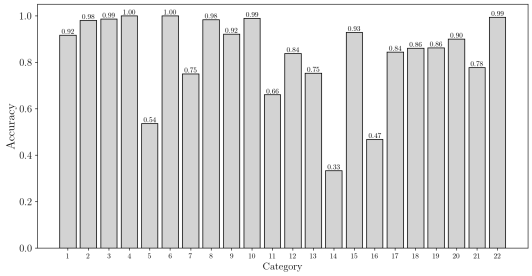
Figure A7a: Algorithm's Accuracy in Different HS Chapters, Chilean Customs Dataset



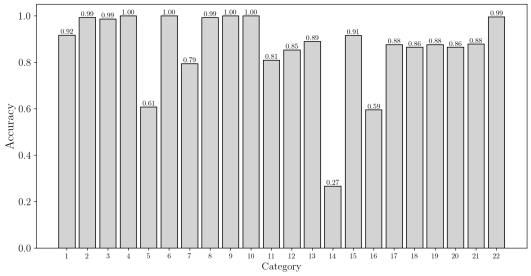
(a) GPT 3.5



(b) GPT 4



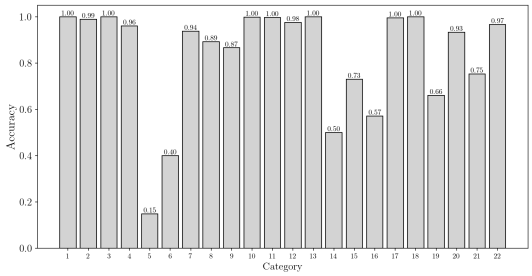
(c) Claude 3 Sonnet



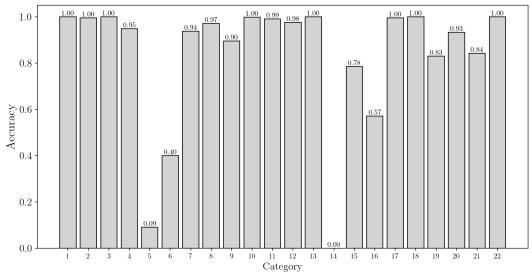
(d) Claude 3.5 Sonnet

Source: Authors' calculations based on Chilean customs data.

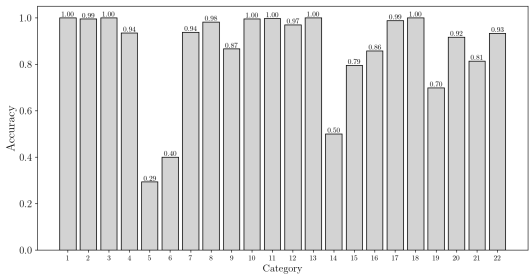
Figure A7b: Algorithm's Accuracy in Different HS Chapters, Paraguayan Customs Dataset



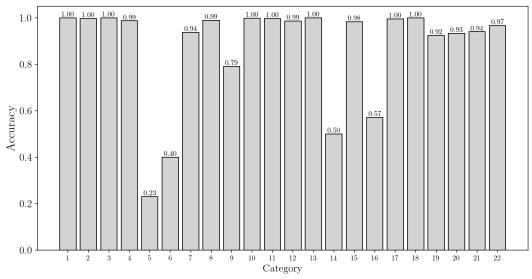
(a) GPT 3.5



(b) GPT 4



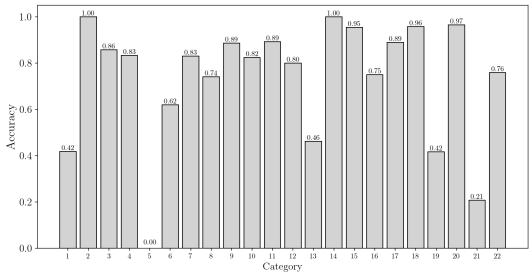
(c) Claude 3 Sonnet



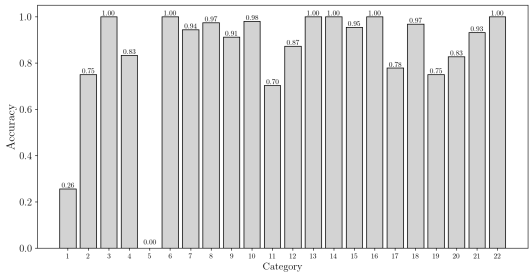
(d) Claude 3.5 Sonnet

Source: Authors' calculations based on Paraguayan customs data.

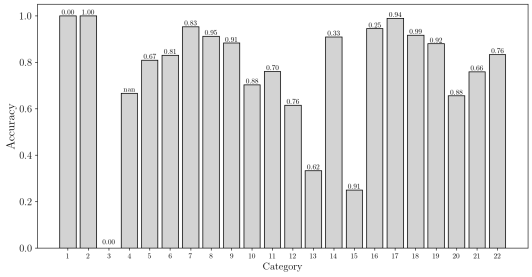
Figure A7c: Algorithm's Accuracy in Different HS Chapters, Organic Dataset



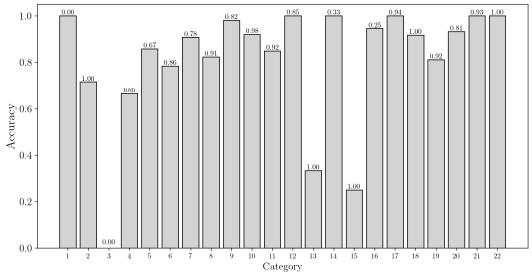
(a) GPT 3.5



(b) GPT 4



(c) Claude 3 Sonnet



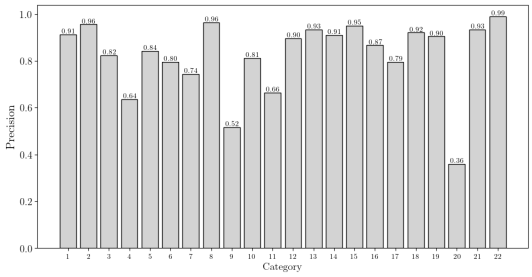
(d) Claude 3.5 Sonnet

Source: Authors' calculations based on USDA data.

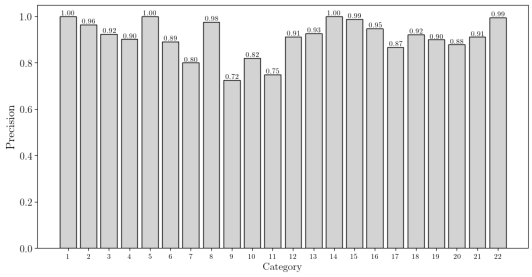
A8 Other Performance Metrics in HS-2 Chapters

In this section, we show the precision, recall, and F1-score of the LLM algorithms at the HS-6 digits for each HS Chapter.

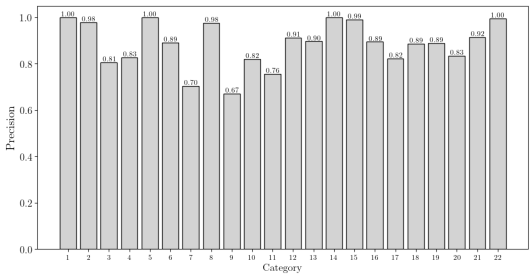
Figure A8a: Algorithm's Precision in Different HS Chapters, Chilean Customs Dataset



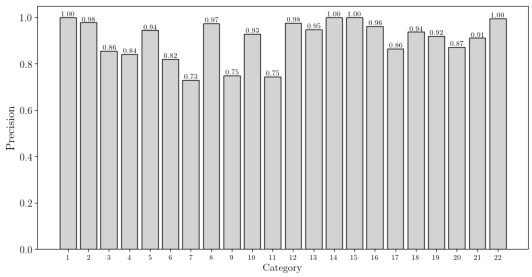
(a) GPT 3.5



(b) GPT 4



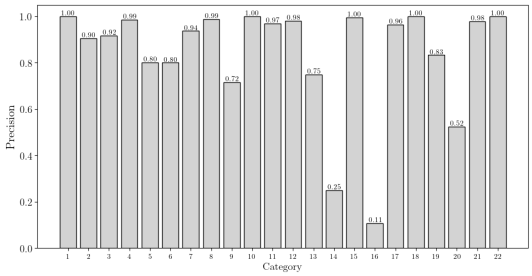
(c) Claude 3 Sonnet



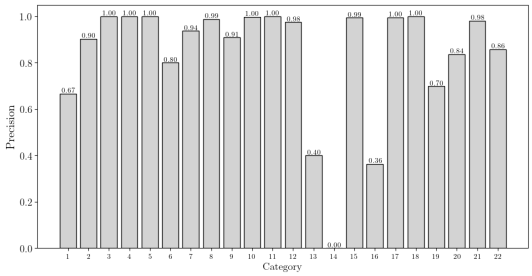
(d) Claude 3.5 Sonnet

Source: Authors' calculations based on Chilean customs data.

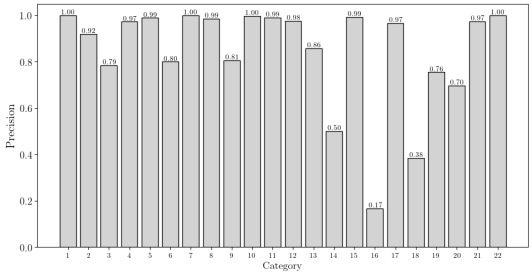
Figure A8b: Algorithm's Precision in Different HS Chapters, Paraguayan Customs Dataset



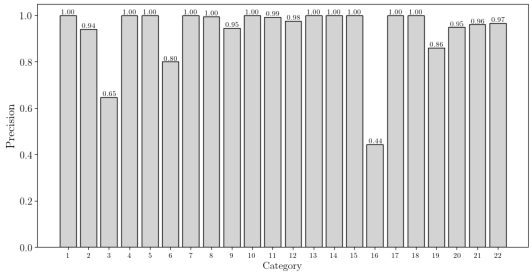
(a) GPT 3.5



(b) GPT 4



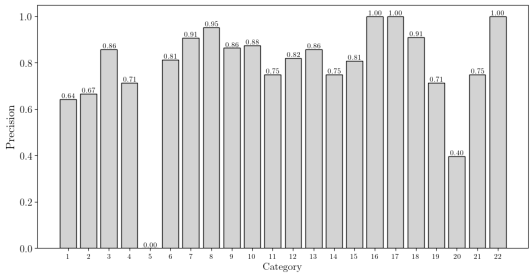
(c) Claude 3 Sonnet



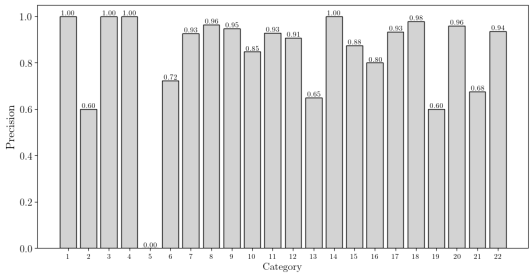
(d) Claude 3.5 Sonnet

Source: Authors' calculations based on Paraguayan customs data.

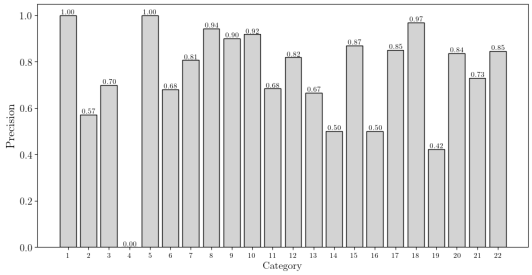
Figure A8c: Algorithm's Precision in Different HS Chapters, Organic Dataset



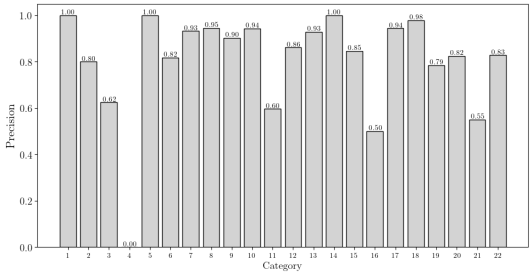
(a) GPT 3.5



(b) GPT 4



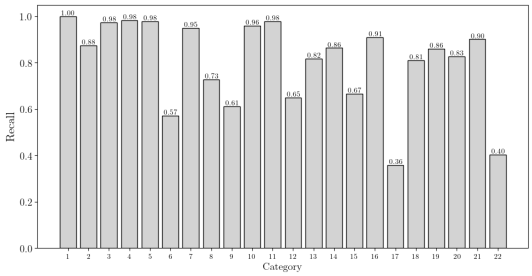
(c) Claude 3 Sonnet



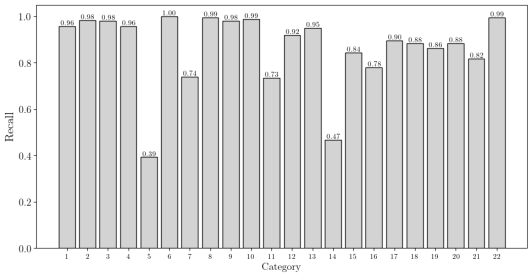
(d) Claude 3.5 Sonnet

Source: Authors' calculations based on USDA data.

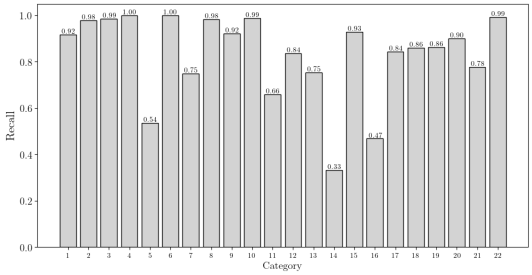
Figure A8d: Algorithm's Recall in Different HS Chapters, Chilean Customs Dataset



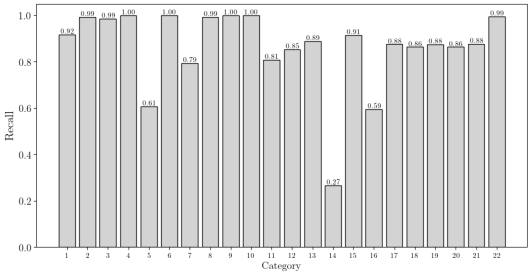
(a) GPT 3.5



(b) GPT 4



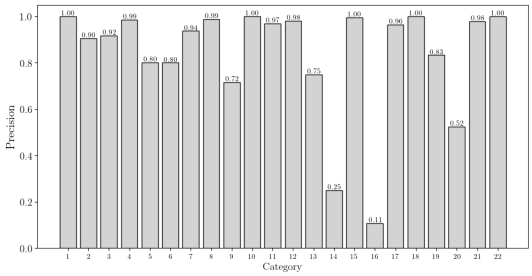
(c) Claude 3 Sonnet



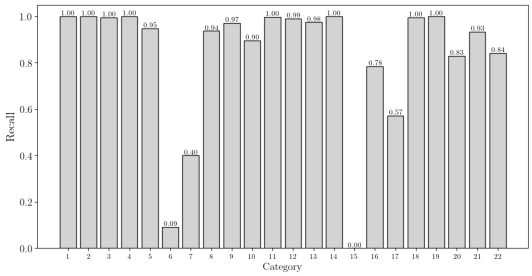
(d) Claude 3.5 Sonnet

Source: Authors' calculations based on Chilean customs data.

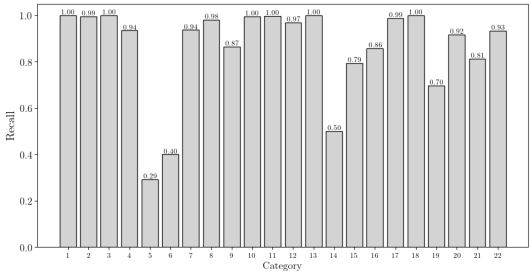
Figure A8e: Algorithm's Recall in Different HS Chapters, Paraguayan Customs Dataset



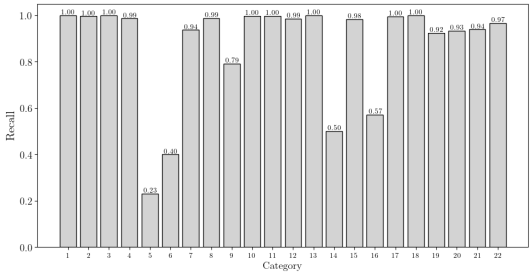
(a) GPT 3.5



(b) GPT 4



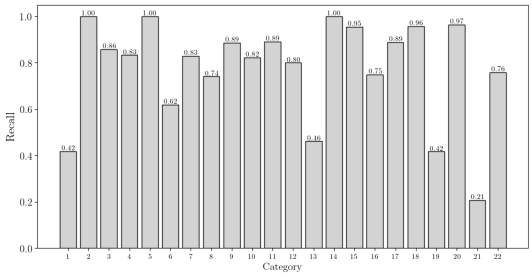
(c) Claude 3 Sonnet



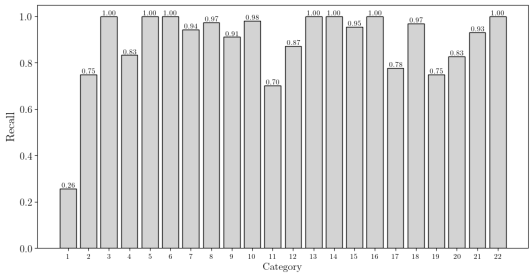
(d) Claude 3.5 Sonnet

Source: Authors' calculations based on Paraguayan customs data.

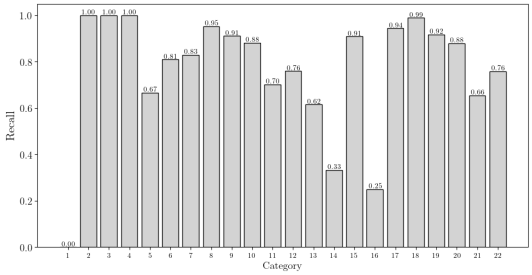
Figure A8f: Algorithm's Recall in Different HS Chapters, Organic Dataset



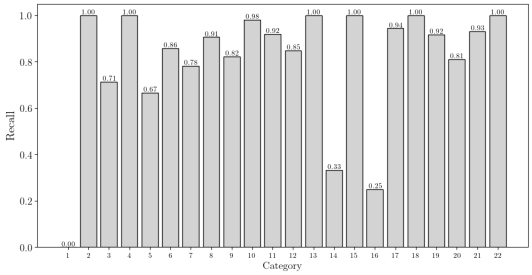
(a) GPT 3.5



(b) GPT 4



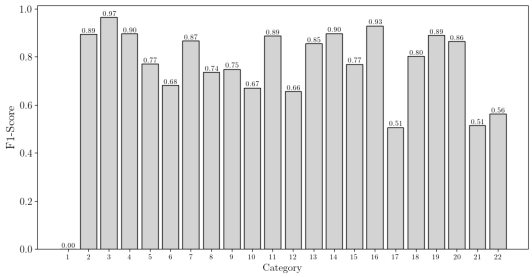
(c) Claude 3 Sonnet



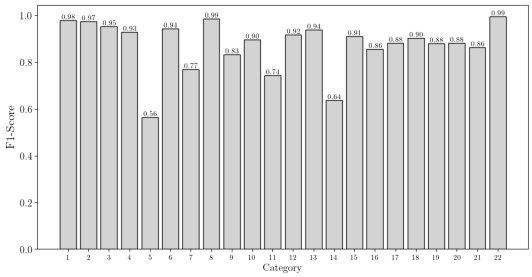
(d) Claude 3.5 Sonnet

Source: Authors' calculations based on USDA data.

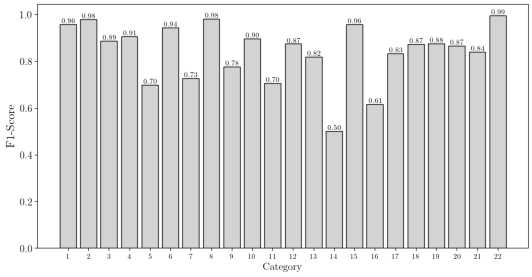
Figure A8g: Algorithm's F1-Score in Different HS Chapters, Chilean Customs Dataset



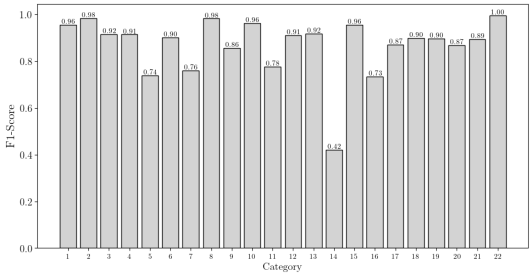
(a) GPT 3.5



(b) GPT 4



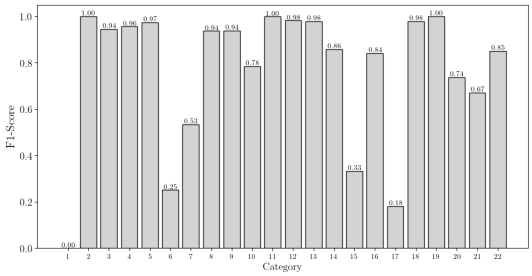
(c) Claude 3 Sonnet



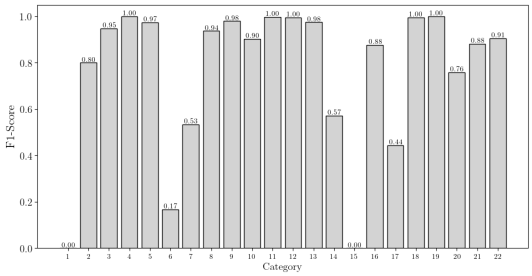
(d) Claude 3.5 Sonnet

Source: Authors' calculations based on Chilean customs data.

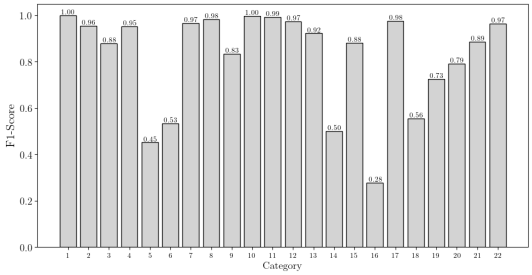
Figure A8h: Algorithm's F1-Score in Different HS Chapters, Paraguayan Customs Dataset



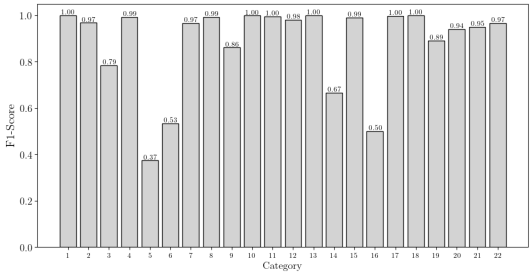
(a) GPT 3.5



(b) GPT 4



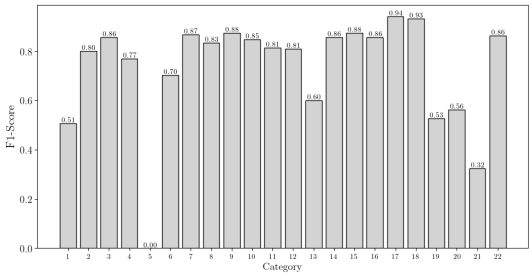
(c) Claude 3 Sonnet



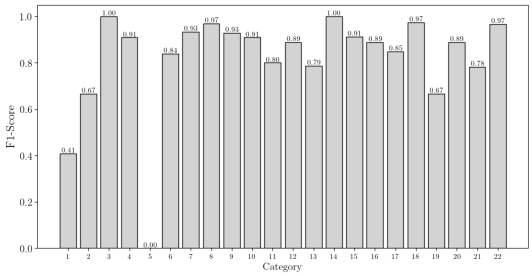
(d) Claude 3.5 Sonnet

Source: Authors' calculations based on Paraguayan customs data.

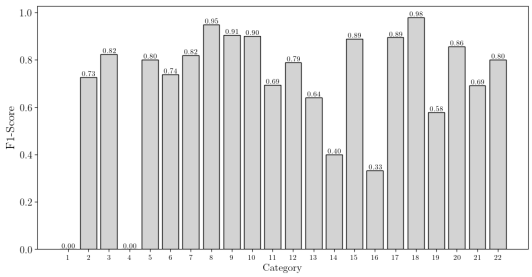
Figure A8i: Algorithm's F1-Score in Different HS Chapters, Organic Dataset



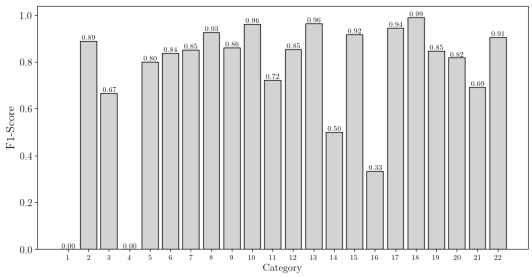
(a) GPT 3.5



(b) GPT 4



(c) Claude 3 Sonnet



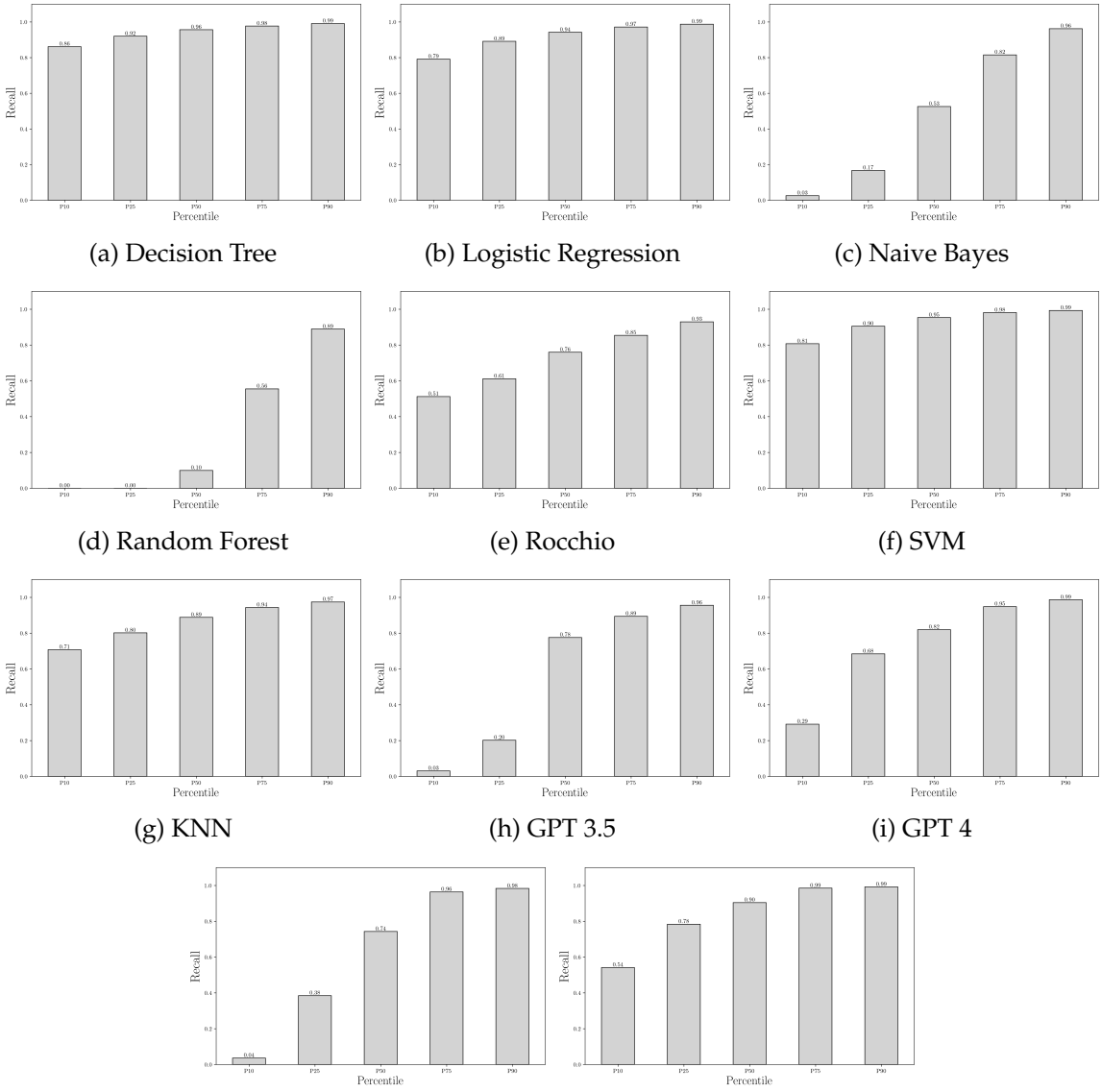
(d) Claude 3.5 Sonnet

Source: Authors' calculations based on USDA data.

A9 Accuracy Distributions for HS4 and HS6 Classification

This section shows the accuracy distributions for more granular classes (HS4 digits and HS6 digits) for each ML and LLM algorithm. Since these classification levels have a large number of classes, the graphs show the different percentiles of the distribution (P10, P25, P50, P75, P90). For instance, P90 corresponds to the accuracy level of the class in the 90th percentile of the distribution.

Figure A9a: Distribution of Accuracy for HS6 Classification by Model in the Chilean Dataset



(a) Decision Tree (b) Logistic Regression (c) Naive Bayes

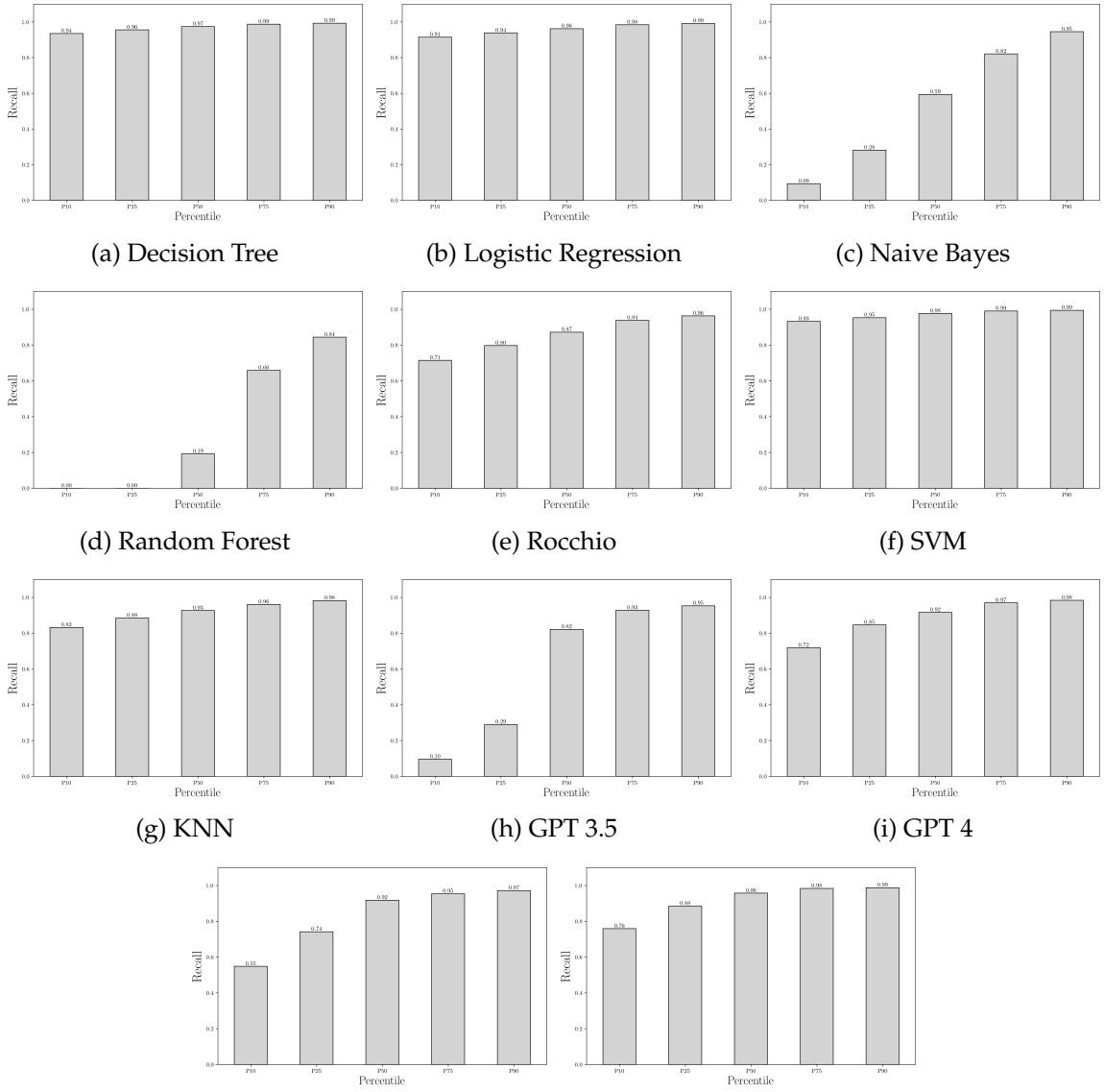
(d) Random Forest (e) Rocchio (f) SVM

(g) KNN (h) GPT 3.5 (i) GPT 4

(j) Claude 3 Sonnet (k) Claude 3.5 Sonnet

Source: Authors' calculations based on Chilean customs data.

Figure A9b: Distribution of Accuracy for HS4 Classification by Model in the Chilean Dataset



(a) Decision Tree (b) Logistic Regression (c) Naive Bayes

(d) Random Forest (e) Rocchio (f) SVM

(g) KNN (h) GPT 3.5 (i) GPT 4

(j) Claude 3 Sonnet (k) Claude 3.5 Sonnet

Source: Authors' calculations based on Chilean customs data.

A10 Training Sample Balanced by HS2 Chapter

In this Appendix, we describe the process followed to balance our training sample for the ML algorithms across HS-2 Chapters. Subsequently, we show the results of the algorithms

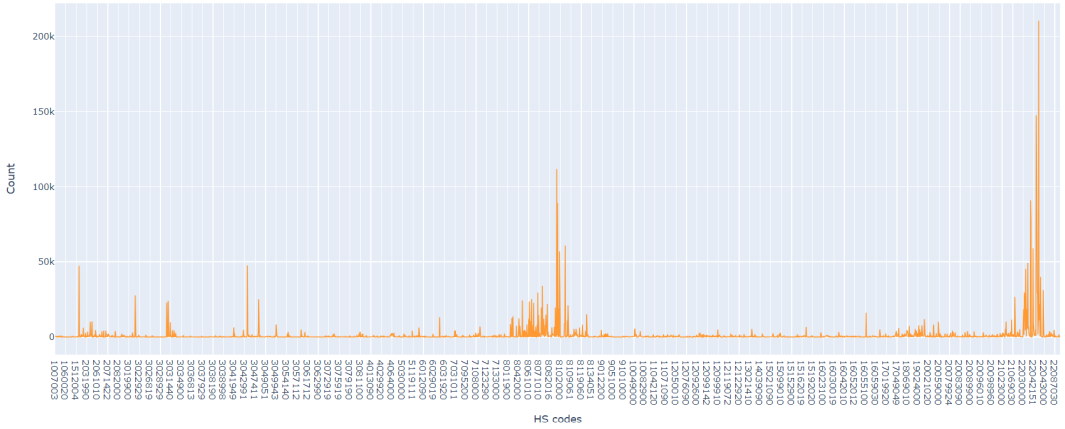
in this alternative balanced sample.

We carry out this robustness check to assess the algorithms in a sample where each aggregate product category (HS2 Chapter) is equally represented in the training dataset. Firstly, we set the number of observations per HS2 Chapter. Approximately 45,000 observations per HS2 are required. This figure is obtained from dividing 1 million observations by the 22 HS2 Chapter. Secondly, we randomize within each HS2 Chapter. For example, out of all the codes available in the database for the first chapter of the HS, 45,271 observations were randomly selected. In this regard, it is very important to consider a characteristic of the original data, which is that not all chapters from 1 to 22 of the HS have more than 45,000 different observations available for selection. Thus, for some chapters, it was necessary to select all the available observations. The final number of observations obtained by HS2 is shown in Table A10a.

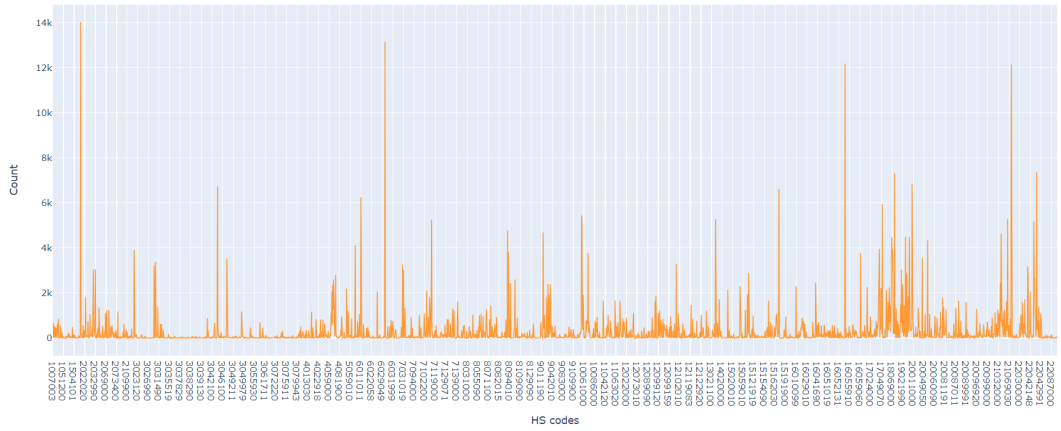
Table A10a: Counting Observations by HS2 in the Balanced and Unbalanced Datasets

HS2	Unbalanced Count	Balanced Count
01	1635	7630
02	63352	45271
03	111537	45148
04	9569	27066
05	4460	13136
06	9486	30482
07	14714	45292
08	255136	45318
09	9480	25383
10	6057	25933
11	5964	15773
12	18879	45336
13	5347	11793
14	1161	3129
15	14002	35559
16	19527	45347
17	11694	26822
18	14783	35157
19	32397	45336
20	34690	45344
21	37849	45401
22	317375	45389
Total	**999094**	**711045**

Figure A10: Observations per HS6 in the unbalanced and balanced models



(a) Unbalanced



(b) Balanced

Source: Authors' calculations based on Chilean customs data.

Table A10b presents the performance of various ML models on the Chilean dataset classified at HS6 level, Table A10c for the Paraguayan dataset and Table A10d for the USDA organic products descriptions.

Table A10b: Classification Report for the Chilean Test-set at HS6 Level

Metric	Decision Tree	Logistic Reg.	Naive Bayes	Random Forest	Rocchio	SVM	KNN	GPT 3.5	GPT Nested	GPT 4
Accuracy	0.94	0.92	0.66	0.54	0.73	0.93	0.86	0.61	0.55	0.75
Macro Avg										
Precision	0.89	0.88	0.94	0.95	0.62	0.91	0.79	0.54	0.43	0.61
Recall	0.85	0.79	0.17	0.12	0.72	0.80	0.62	0.40	0.46	0.51
F1-Score	0.85	0.80	0.19	0.12	0.62	0.82	0.64	0.13	0.09	0.29
Weighted Avg										
Precision	0.94	0.92	0.79	0.76	0.78	0.93	0.86	0.79	0.74	0.83
Recall	0.94	0.92	0.66	0.54	0.73	0.93	0.86	0.61	0.55	0.75
F1-Score	0.94	0.92	0.62	0.48	0.74	0.92	0.86	0.59	0.55	0.73

Source: Authors' calculations based on Chilean customs data.

Table A10c: Classification Report for the Paraguay Dataset at HS6 Level

Metric	Decision Tree	Logistic Reg.	Naive Bayes	Random Forest	Rocchio	SVM	KNN	GPT 3.5	GPT Nested	GPT 4
Accuracy	0.15	0.17	0.08	0.06	0.19	0.20	0.18	0.60	0.46	0.74
Macro Avg										
Precision	0.38	0.38	0.73	0.77	0.43	0.40	0.38	0.42	0.27	0.50
Recall	0.63	0.62	0.29	0.25	0.61	0.61	0.63	0.62	0.73	0.68
F1-Score	0.08	0.09	0.05	0.05	0.12	0.10	0.10	0.14	0.11	0.28
Weighted Avg										
Precision	0.93	0.95	0.96	0.97	0.95	0.95	0.92	0.83	0.76	0.92
Recall	0.15	0.17	0.08	0.06	0.19	0.20	0.18	0.60	0.46	0.74
F1-Score	0.16	0.17	0.07	0.05	0.21	0.19	0.19	0.59	0.49	0.76

Source: Authors' calculations based on Paraguayan customs data.

Table A10d: Classification Report for the Organic Products Dataset at HS6

Metric	Decision Tree	Logistic Reg.	Naive Bayes	Random Forest	Rocchio	SVM	KNN	GPT 3.5	GPT Nested	GPT 4
Accuracy	0.11	0.10	0.08	0.09	0.10	0.11	0.10	0.73	0.30	0.66
Macro Avg										
Precision	0.59	0.46	0.80	0.89	0.55	0.48	0.47	0.71	0.40	0.66
Recall	0.30	0.39	0.15	0.10	0.36	0.38	0.39	0.72	0.57	0.77
F1-Score	0.08	0.06	0.04	0.03	0.07	0.06	0.06	0.49	0.13	0.51
Weighted Avg										
Precision	0.64	0.56	0.80	0.90	0.63	0.58	0.56	0.88	0.68	0.84
Recall	0.11	0.10	0.08	0.09	0.10	0.11	0.10	0.73	0.30	0.66
F1-Score	0.09	0.09	0.07	0.06	0.11	0.11	0.10	0.73	0.32	0.66

Source: Authors' calculations based on USDA.

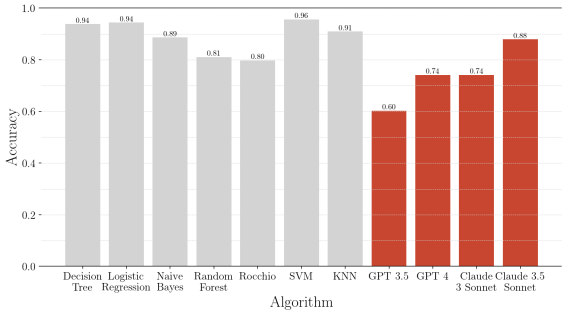
A11 Alternative Train-Test Dataset: Training with Paraguayan Data

In this Appendix, we switch the test-train dataset and thus train the ML models with Paraguayan data, testing the newly trained models on the Chilean and USDA organic databases. This ex-

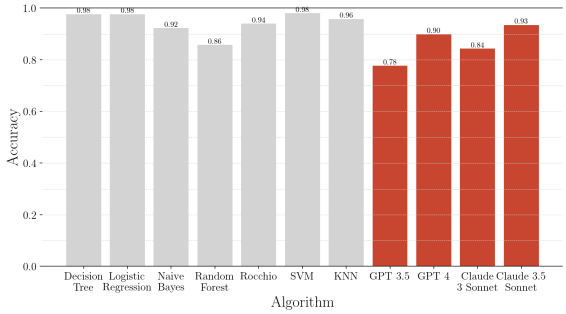
ercise was carried out as a robustness test to assess the generalizability of our conclusions regarding (i) the dramatic drop in performance of ML algorithms when assessed outside the train-test-split dataset and (ii) the high relative performance of LLMs in these external datasets.

As before, the results are presented at three Harmonized System (HS) classification levels: HS-6, HS-4, and HS-2. Figure A11a illustrates the algorithms' accuracy in the Paraguayan dataset (the test-train database for this exercise). Figure A11b depicts the algorithms' accuracy when applied to the Chilean dataset (first external dataset on this exercise). As anticipated, there is a general decrease in accuracy compared to the Paraguayan results. This reduction is more pronounced at the HS-6 level, suggesting that maintaining a high accuracy across datasets is more challenging at more granular classification levels. Figure A11c shows similar results for the Organic dataset, with a further decrease in accuracy.

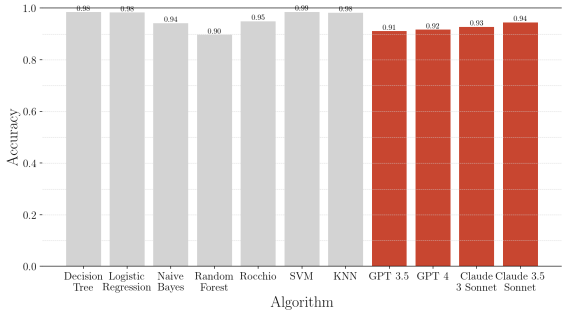
Figure A11a: Algorithm's Accuracy in the Test-Train-Split Dataset: Paraguayan Customs.



(a) HS-6 digit level



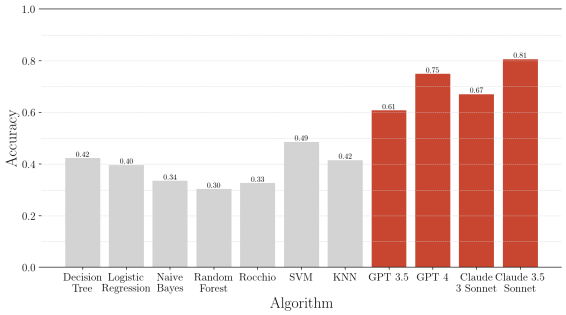
(b) HS-4 digit level



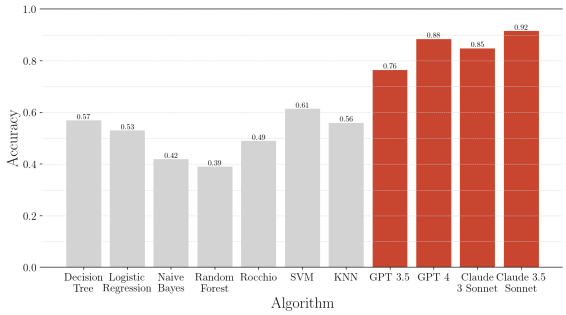
(c) HS-2 digit level

Source: Authors' calculations based on Paraguayan customs data.

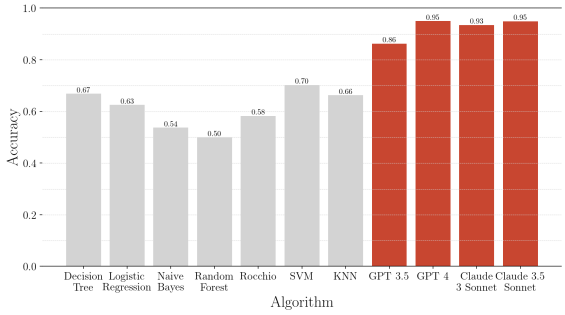
Figure A11b: Algorithm's Accuracy in the Chilean Dataset.



(a) HS-6 digit level



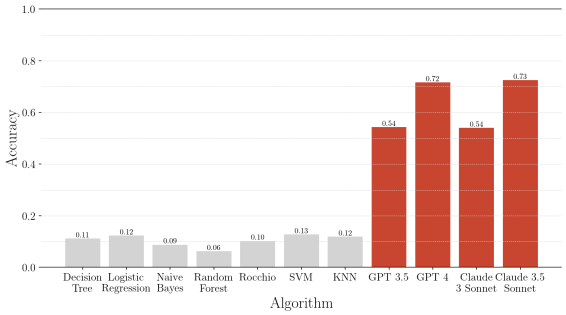
(b) HS-4 digit level



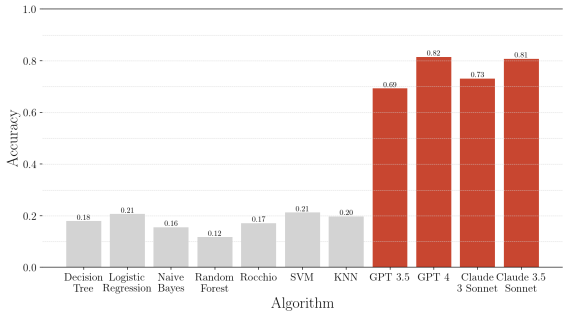
(c) HS-2 digit level

Source: Authors' calculations based on Chilean customs data.

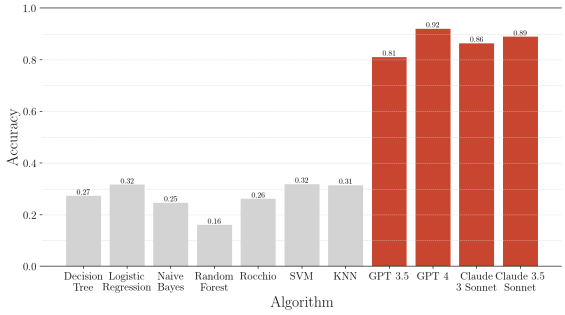
Figure A11c: Algorithm's Accuracy in the Organic Dataset.



(a) HS-6 digit level



(b) HS-4 digit level



(c) HS-2 digit level

Source: Authors' calculations based on USDA.

Tables A11a, A11b, and A11c provide more detailed multiclass classification performance metrics for the three datasets at the HS-6 level.

Table A11a: Classification Report for the Paraguayan Test-Set at HS6 Level

Metric	Dec. Tree	Log. Reg.	Naive Bayes	Rand. Forest	Ro-cchio	SVM	KNN	GPT 3.5	GPT Nested	GPT 4	Claude 3 S	Claude 3.5 S
Accuracy	0.94	0.94	0.89	0.81	0.80	0.96	0.91	0.60	0.46	0.74	0.74	0.88
Macro Avg												
Precision	0.83	0.83	0.87	0.88	0.60	0.87	0.80	0.42	0.27	0.50	0.46	0.56
Recall	0.88	0.85	0.41	0.30	0.76	0.86	0.82	0.62	0.73	0.68	0.69	0.75
F1-Score	0.75	0.73	0.35	0.24	0.49	0.78	0.69	0.14	0.11	0.28	0.23	0.38
Weighted Avg												
Precision	0.97	0.97	0.92	0.88	0.95	0.97	0.94	0.83	0.76	0.92	0.90	0.95
Recall	0.94	0.94	0.89	0.81	0.80	0.96	0.91	0.60	0.46	0.74	0.74	0.88
F1-Score	0.95	0.95	0.87	0.78	0.86	0.96	0.92	0.59	0.49	0.76	0.73	0.87

Source: Authors' calculations based on Paraguayan customs data.

Table A11b: Classification Report for the Chilean Dataset at HS6 Level

Metric	Dec. Tree	Log. Reg.	Naive Bayes	Rand. Forest	Rocchio	SVM	KNN	GPT 3.5	GPT Nested	GPT 4	Claude 3 S	Claude 3.5 S
Accuracy	0.42	0.40	0.34	0.30	0.33	0.49	0.42	0.61	0.55	0.75	0.67	0.81
Macro Avg												
Precision	0.55	0.52	0.89	0.93	0.54	0.56	0.54	0.54	0.43	0.61	0.58	0.64
Recall	0.22	0.23	0.08	0.06	0.24	0.27	0.25	0.40	0.46	0.51	0.47	0.60
F1-Score	0.15	0.15	0.06	0.05	0.17	0.18	0.16	0.13	0.09	0.29	0.20	0.38
Weighted Avg												
Precision	0.58	0.55	0.73	0.80	0.58	0.61	0.57	0.79	0.74	0.83	0.84	0.86
Recall	0.42	0.40	0.34	0.30	0.33	0.49	0.42	0.61	0.55	0.75	0.67	0.81
F1-Score	0.41	0.39	0.28	0.25	0.34	0.47	0.41	0.59	0.55	0.73	0.66	0.79

Source: Authors' calculations based on Chilean customs data.

Table A11c: Classification Report for the Organic Dataset at HS6 Level

Metric	Dec. Tree	Log. Reg.	Naive Bayes	Rand. Forest	Rocchio	SVM	KNN	GPT 3.5	GPT Nested	GPT 4	Claude 3 S	Claude 3.5 S
Accuracy	0.11	0.12	0.08	0.06	0.10	0.12	0.11	0.54	0.28	0.72	0.54	0.73
Macro Avg												
Precision	0.60	0.49	0.75	0.92	0.63	0.53	0.53	0.64	0.43	0.69	0.59	0.73
Recall	0.28	0.34	0.15	0.07	0.27	0.32	0.31	0.58	0.54	0.66	0.63	0.73
F1-Score	0.07	0.06	0.04	0.03	0.07	0.06	0.06	0.33	0.13	0.45	0.33	0.55
Weighted Avg												
Precision	0.65	0.64	0.77	0.88	0.71	0.67	0.60	0.80	0.68	0.84	0.77	0.84
Recall	0.11	0.12	0.08	0.06	0.10	0.12	0.11	0.54	0.28	0.72	0.54	0.73
F1-Score	0.11	0.13	0.07	0.04	0.10	0.13	0.12	0.53	0.31	0.71	0.54	0.72

Source: Authors' calculations based on USDA.

A12 LLM Nested Prompt

In this Appendix, we describe LLM Nested Prompting and summarize the results from this alternative prompting strategy. The process consists of assigning codes in a progressive manner, starting with HS2, then HS4 within that HS2, and finally HS6 within that HS4. In this process, the code assigned in the previous stage is indicated so that the model has this information when assigning a new code. In other words, we seek that within the codes assigned at a higher level, the model looks for the next code that most closely resembles the description provided.

```
@backoff.on_exception(backoff.expo, (openai.error.RateLimitError, requests.
    exceptions.ReadTimeout), max_time=60)
def asignar_codigo_HS2(fila, columna_descripcion):
    texto = fila[columna_descripcion]
    modelo = "gpt-3.5-turbo"
    response = openai.ChatCompletion.create(
        model=modelo,
        messages=[
                    {"role": "system", "content": "You are a helpful assistant that
                         assign codes in the Harmonized System of UN Comtrade."},
                    {"role": "user", "content": f'Please assign the harmonized
                        system code number in the HS2 for the following description
                        :"{texto}". Return "Code: number here". If you are unsure
                        of the classification, provide your best possible option'}
                ],
        temperature = 0.1
    )
    codigo_asignado = response['choices'][0]['message']['content']
    return codigo_asignado
```

```
@backoff.on_exception(backoff.expo, (openai.error.RateLimitError, requests.
    exceptions.ReadTimeout), max_time=60)
def asignar_codigo_HS4(fila, columna_descripcion, columna_codigo_anterior):
    texto = fila[columna_descripcion]
    codigo_anterior = fila[columna_codigo_anterior]
    modelo = "gpt-3.5-turbo"
    response = openai.ChatCompletion.create(
        model=modelo,
        messages=[
                    {"role": "system", "content": "You are a helpful assistant that
                         assign codes in the Harmonized System of UN Comtrade."},
```

```
                    {"role": "user", "content": f'Please assign the harmonized
                        system code number in the HS4 for the following description
                        :"{texto}". Bear in mind that you gave me the next HS2 Code
                        : {codigo_anterior} and that the HS4 Code has to be inside
                        the HS2. Return "Code: number here". If you are unsure of
                        the classification, provide your best possible option'}
                ],
        temperature = 0.1
    )
    codigo_asignado = response['choices'][0]['message']['content']
    return codigo_asignado
```

```
@backoff.on_exception(backoff.expo, (openai.error.RateLimitError, requests.
    exceptions.ReadTimeout), max_time=60)
def asignar_codigo_HS6(fila, columna_descripcion, columna_codigo_anterior):
    texto = fila[columna_descripcion]
    codigo_anterior = fila[columna_codigo_anterior]
    modelo = "gpt-3.5-turbo"
    response = openai.ChatCompletion.create(
        model=modelo,
        messages=[
                    {"role": "system", "content": "You are a helpful assistant that
                         assign codes in the Harmonized System of UN Comtrade."},
                    {"role": "user", "content": f'Please assign the harmonized
                        system code number in the HS6 for the following description
                        :"{texto}". Bear in mind that you gave me the next HS4 Code
                        : {codigo_anterior} and that the HS6 Code has to be inside
                        the HS4. Return "Code: number here". If you are unsure of
                        the classification, provide your best possible option'}
                ],
        temperature = 0.1
    )
    codigo_asignado = response['choices'][0]['message']['content']
    return codigo_asignado
```

Table A12a: GPT Nested Classification Results

HS Level	Metric	Chile	Paraguay	Organic
HS6	Accuracy	0.55	0.46	0.28
	Macro Avg			
	Precision	0.43	0.27	0.43
	Recall	0.46	0.73	0.54
	F1-Score	0.09	0.11	0.13
	Weighted Avg			
	Precision	0.74	0.76	0.68
	Recall	0.55	0.46	0.28
	F1-Score	0.55	0.49	0.31
HS4	Accuracy	0.78	0.71	0.53
	Macro Avg			
	Precision	0.41	0.32	0.51
	Recall	0.73	0.83	0.64
	F1-Score	0.31	0.25	0.38
	Weighted Avg			
	Precision	0.85	0.77	0.70
	Recall	0.78	0.71	0.53
	F1-Score	0.78	0.69	0.57
HS2	Accuracy	0.90	0.86	0.69
	Macro Avg			
	Precision	0.20	0.20	0.49
	Recall	0.94	0.95	0.78
	F1-Score	0.18	0.19	0.42
	Weighted Avg			
	Precision	0.92	0.91	0.82
	Recall	0.90	0.86	0.69
	F1-Score	0.90	0.84	0.74

Source: Authors' calculations.

A13 Exploring Other LLMs

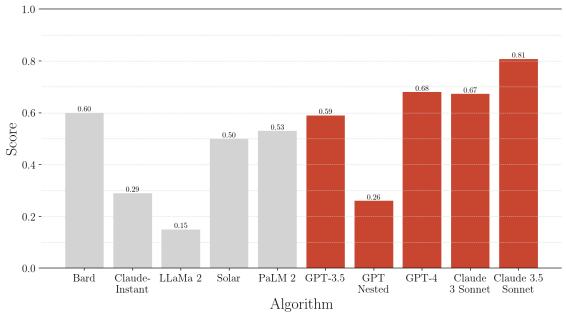
In this section, we present a benchmark of other LLMs tasked with classifying products from text descriptions. For this task, we selected 100 random observations from our test dataset with data from the Chilean customs (see Section 2.1 for more details). This exercise is carried out through Poe, an open platform to explore LLMs. As of November 2023, this service is not accessible at scale through an API, preventing us from carrying out a more extensive analysis of these alternative LLMs. However, note that, while a random sample of 100 observations is relatively small, in Appendix A14, we show that our point estimates for GPT 3.5 and GPT 4 are very stable when moving from 100 to 1000 observations.

Figures A13a illustrates the accuracy rates of five distinct LLMs at HS 6-digit classifications: Bard[30], Claude-Instant[31], LLaMa 2[32], Solar[33], PaLM 2, GPT 3.5, and GPT 4. Each bar represents the algorithm's product classification efficacy. Both Bard, PaLM 2 and Solar exhibit relatively high accuracy rates, with Bard edging ahead slightly (60%). PaLM 2 demonstrates a robust performance (53%), aligning closely with Solar's (50%). GPT 4 emerges as the leading algorithm, with a score substantially higher than the others.

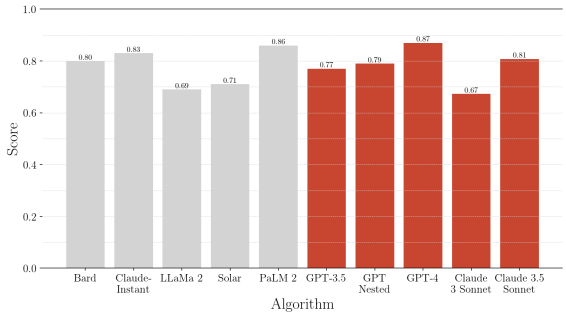
When we move to lower levels of disaggregation in Figures A13b (HS 4-digits) and A13c (HS 2-digits), the differences relative to GPT 4 become smaller. While GPT 4 remains the leading algorithm, Palm 2 (86%) is only 1 percentage point behind at the HS 4-digit level and show equal precision at HS 2-digit level (92%). All algorithms perform well at the HS 2-digit level, indicating a remarkable increase in accuracy in more aggregate categorizations. LlaMa 2, for example, assigns the correct HS 6 digit-code only in 15% of the cases, but its accuracy surges to 69% at HS 4-digit level and to 83% at the HS 2- digit level.

[30] Bard AI is a conversational AI chatbot developed by Google AI. It is powered by PaLM 2, a 540-billion parameter model, created by Google Research and trained with the Pathways system.
[31] Claude is an LLM developed by Anthropic with roughly 175 billion parameters.
[32] LlaMa 2 is a family of almost open-source LLMs (excluding commercial use). Here we used the 70-billion parameter version.
[33] Solar-0-70b-16bit is a fine-tuned version of LlaMa 2 and a top-ranked model on the HuggingFace Open LLM leaderboard.

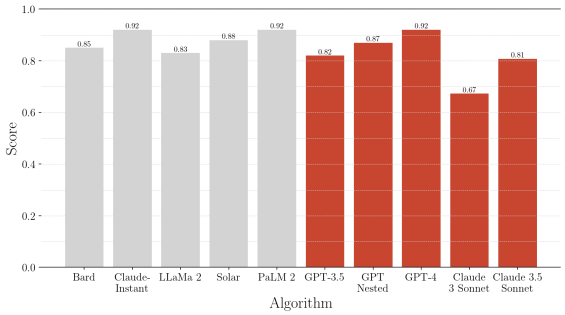
Figure A13: Comparative Performance of Other LLMs in HS Product Classification



(a) HS-6 level



(b) HS-4 level



(c) HS-2 level

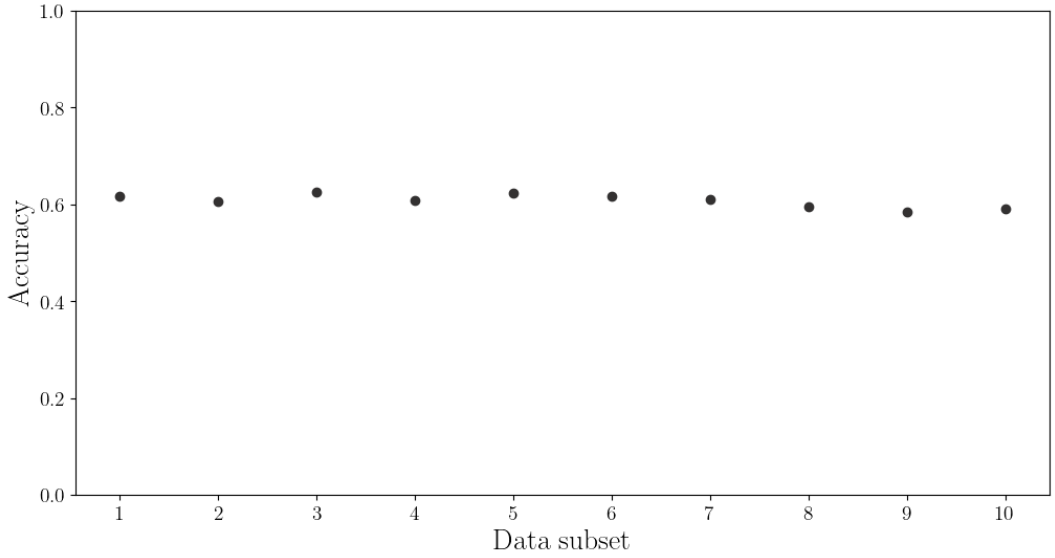
Source: Authors' calculations based on Chilean customs data.

A14 LLM Subsample Accuracy

In this section, we report results of robustness checks of our findings to a reduction in the sample size by one order of magnitude. In particular, we randomly divide a 1000-observations sample into 10 subsamples of 100 observations and test the accuracy of GPT 3.5 and GPT 4 in each subsample for each dataset.

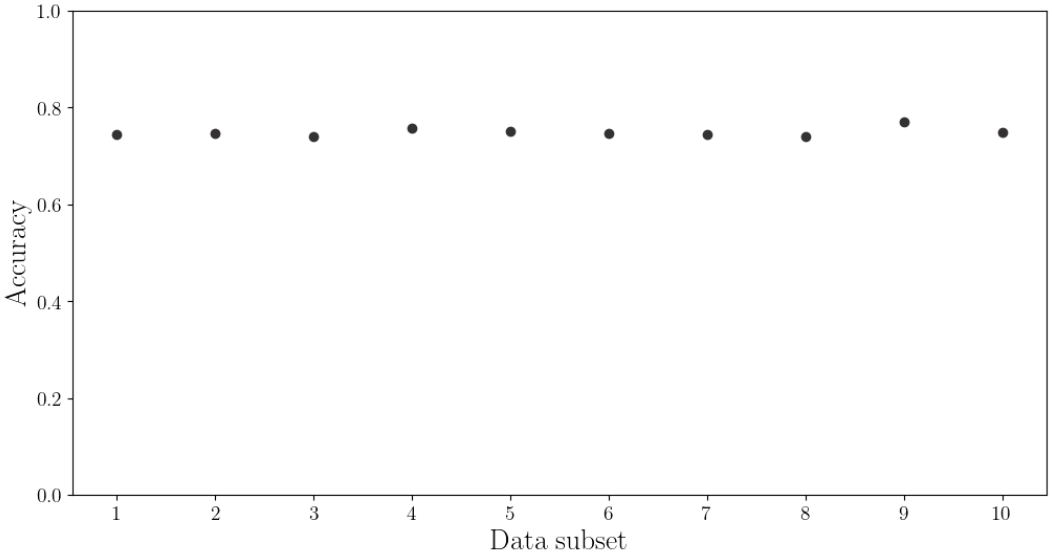
Figure A14a and A14b shows the point estimates after dividing the sample of observations from Chile into 10 groups. In this case, the point estimates across the 10 subsamples have a very small standard deviation of 0.0048 for GPT 3.5 and 0.0043 for GPT 4. Figure A14c and A14d does the same for the Paraguayan dataset. The standard deviation in this case is very similar (0.0049 and 0.0044 for GPT 3.5 and GPT 4, respectively). Finally, Figure A14e and A14f does the same for the 1,000 classified observations from the USDA organic product database, for which the standard deviation is slightly larger, at 0.0136 for GPT 3.5 and 0.0149 for GPT 4. Overall, this exercise shows that the point coefficients are relatively stable when moving from a 1000-observations sample to 100-observations random subsamples.

Figure A14a: Algorithm's Accuracy in Chilean Subsets for GPT 3.5



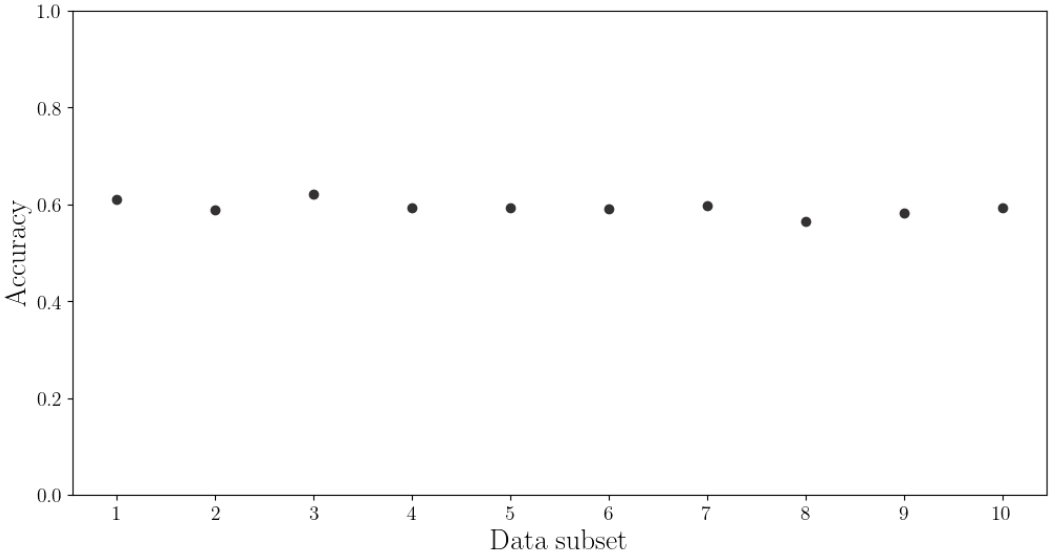
Source: Authors' calculations based on data from Chilean Customs.

Figure A14b: Algorithm's Accuracy in Chilean Subsets for GPT 4



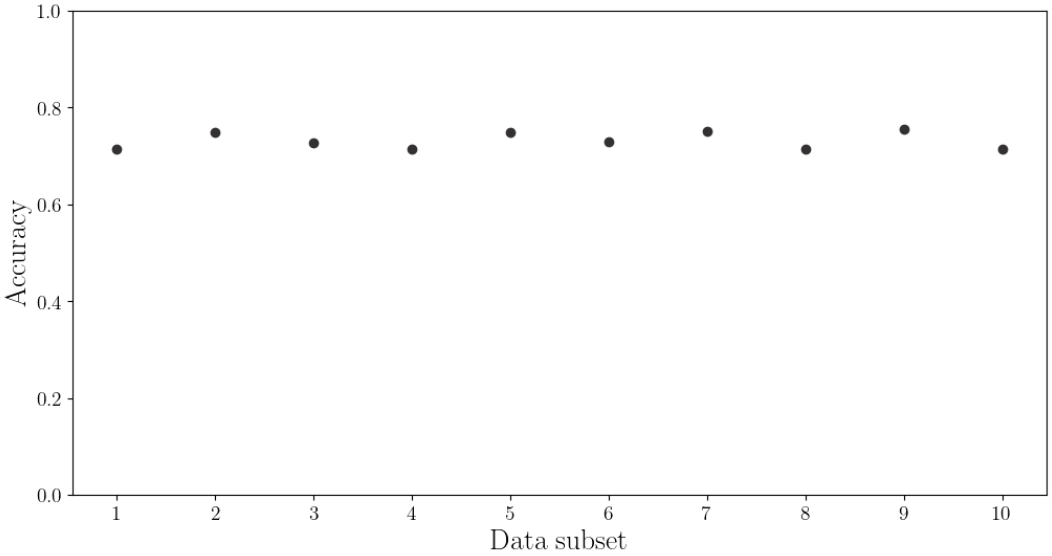
Source: Authors' calculations based on data from Chilean Customs.

Figure A14c: Algorithm's Accuracy in Paraguayan Subsets for GPT 3.5



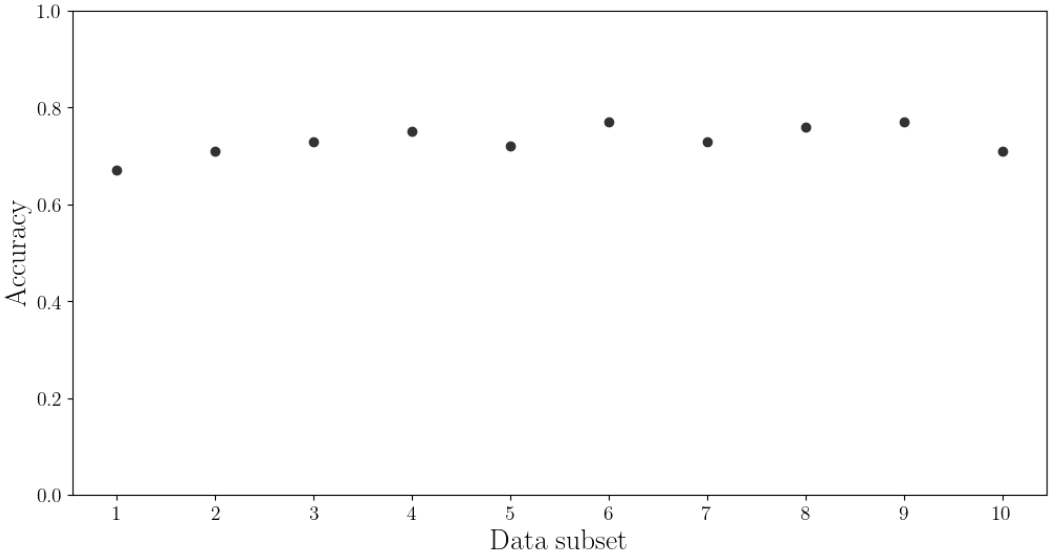
Source: Authors' calculations based on data from Paraguayan Customs.

Figure A14d: Algorithm's Accuracy in Paraguayan Subsets for GPT 4



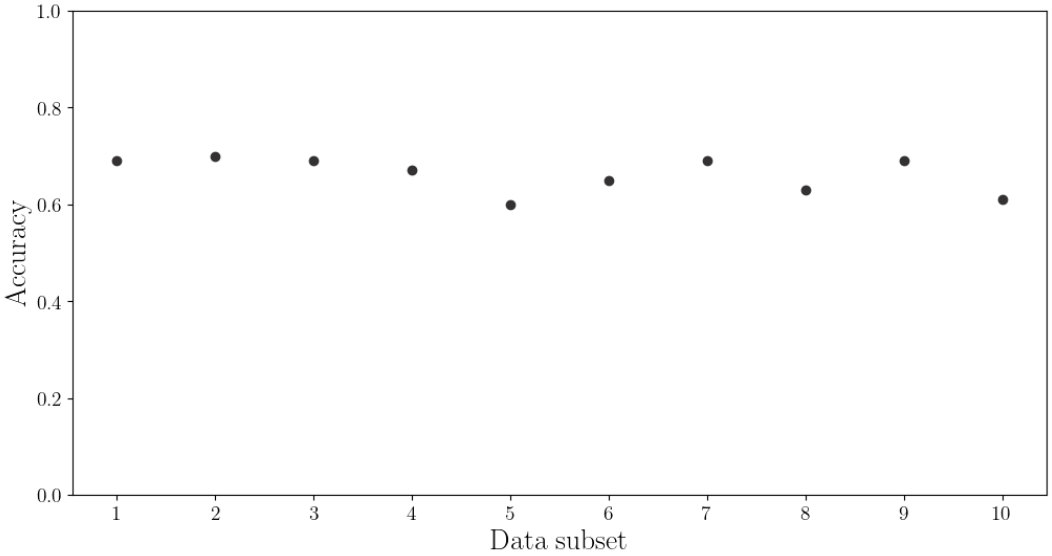
Source: Authors' calculations based on data from Paraguayan Customs.

Figure A14e: Algorithm's Accuracy in the USDA Organic Subsets for GPT 3.5



Source: Authors' calculations based on data from USDA.

Figure A14f: Algorithm's Accuracy in the USDA Organic Subsets for GPT 4



Source: Authors' calculations based on data from USDA.