Identifying and mitigating misclassification: A case study of the Machine Learning lifecycle in price indices with web-scraped clothing data

William Spackman, Greg DeVilliers, Christian Ritter, Serge Goussev

Abstract:

While the application of Supervised Machine Learning (ML) to automate the classification of alternative data for official price indices has been widely demonstrated, the impact of misclassification within the ML lifecycle, from initial annotation of the training data to retraining models due to data drift, has been understudied in the literature. To support National Statistical Offices in understanding how to apply ML to support at-scale production needs, our research provides an empirical case study of how misclassification could be present at major stages of a ML lifecycle, its impact on elementary price indices and ways it can be mitigated through model retraining or validation processes.

Keywords: Price Indices, Machine Learning, Misclassification, Quality Assurance

1. Introduction

National Statistical Offices (NSOs) have increasingly turned to alternative data sources (point of sale or transaction data, web-scraped data, and administrative data) to augment traditional field-collected data in the compilation of official price indices such as the Consumer Prices Index (CPI). To utilize such large datasets in production, Machine Learning (ML) has been widely investigated for the critical task of classification (Myklatun 2019; Harms and Spinder 2019; Office for National Statistics 2020) – the categorization of unique products available in a retailer's dataset to the lowest level of a classification taxonomy utilized by the NSO. As classification is as an interim step applied prior to aggregation, any classification errors could lead to measurement error within final statistics if no quality control process is in place to correct potential errors and validate the classified data (Yung, et al. 2020; Scholtus and van Delden 2020; Meertens, Van den Herik and Takes 2020). While methodological literature has demonstrated that misclassification could affect statistical outputs such as counts or total turnover (Scholtus and van Delden 2020), we are not aware of a detailed and comprehensive discussion of the impacts of misclassification on the price indices, specifically when applying Machine Learning on alternative data. A consideration of the topic is critical as NSOs design at-scale classification that balances the cost of quality control with impact from possible classification errors on price indices.

The objective of this study is to introduce the topic of misclassification errors within the alternative data sources to the price indices literature. We provide an empirical case study on key aspects that NSOs consider when applying Machine Learning for production. These aspects include (a) data labelling (or annotation) patterns that are important to consider when creating representative labelled datasets for ML model training or validation of data in production; (b) evaluation of how misclassification could impact the elementary indices: the building blocks of the CPI; (c) ML model decay over time; (d) and outlier detection strategies to flag products for manual review in order to improve ML model performance. While not exclusive of all considerations NSOs face when applying ML to alternative data, these research areas address the foundational aspects that support other considerations and aid further research on the impact of misclassification errors. The goal of the paper is thus to provide an overview of

the impact of misclassification across the key phases rather than dive deeply into each sub-topic, which is left for later research.

Conceptually, misclassification could impact price indices when the principle of homogeneity in elementary aggregates, or similarity in characteristics, content, or price change, is affected (Manual, Consumer Price Index 2020). Specifically, if a large proportion of products are wrongly classified into a category and have a price movement tendency different than the correctly classified products in that category, then over time, the index for that category would show an incorrect price movement. In real-world situations, homogeneity may not be a criterion that is fully matched. Furthermore, heterogeneity in the domain-specific natural language to describe products in the category increases the difficulty for an ML model to generalize and naturally increases misclassification. Understanding how misclassification affects elementary aggregates is key as it guides subsequent steps. Maintenance of quality is central, justifying the effort of NSOs to design processes (HLG MOS 2019; Yung, et al. 2020), as well as undertaken research to mitigate the impact of misclassification on statistical outputs (Oyarzun and Wile 2022). Manual validation has become a standard approach applied on newly classified records prior to using these datasets for production. This has similarly been the case in price indices, where manual validation of a high proportion of new records is a standard recommendation (Eurostat 2017).

While manual validation is recognized, an additional aspect critical for NSOs applying ML is to assess is how stable model performance is over time, as this will impact how to design the quality control process. Specifically, inherent in practical applications of ML is the likelihood of dataset shift (also often referred to as drift, which describes changes to the data distribution, for example changes of product descriptions and classification) over time in real-world applications. When dataset shift occurs, the assumption that training and production datasets follow the same distributions (independent and identically distributed) is invalidated (Moreno-Torres, et al. 2012). A classifier tested on an original dataset prior to deployment into production is thus unlikely to perform at the same level once a shift has occurred, causing additional misclassifications over time (Scholtus and van Delden 2020, 18). As alternative data sources are not originally intended for statistical output, they are naturally likely to change over time (such as a retailer changing product descriptions on its website, or the prevalence of specific products naturally changing over time due to evolution of consumer preferences), thus affecting classification and subsequent measurement. Presence of shift over time reinforces the need to designing quality control processes, as validating new products each period creates a new ground truth dataset that could be utilized for model monitoring and retraining as necessary: a topic that has attracted considerable focus for both NSOs (Piela 2021; 2022), as well as within the larger Data Science discipline within the topic of ML Operations (MLOps) (Sculley, et al. 2015; Huyen 2022; Valliappa Lakshmanan 2020). Monitoring model errors and re-training models is an optimal approach to address model degradation and maintain quality in statistical outputs, compared to monitoring data-distribution/covariate drift or fixed schedules (Choi, et al. 2022). Retraining models can be done using approaches to update models (either single via online learner and forgetting mechanism, or ensembles, both of which are updated with new data) or train models from scratch (Gama 2013).

Manual validation, akin to annotating or labelling initial datasets used to train supervised ML models, is a nontrivial task. Retailer datasets where ML is often applicable typically do not contain variables that would support simple and robust automated labelling and thus depend on annotators to label unique records to the taxonomy utilized within the NSO. As such, there is no reference corpus, and correctness depends on the process that is designed (Artstein 2017). This is particularly challenging given that the lowest levels of taxonomies to which classification needs to be done are at times heterogeneous and subjective (Greenhough, Martindale and Sands 2022). Understanding the subjectivity and heterogeneity of the categories and of the products in each retailer is thus key for NSOs to design manual annotation or validation processes, as it is impractical to allocate multiple annotators (or validators) to all categories equally: some may not need the extra investment, while it is critical for

others. Design of a robust validation process furthermore acts as a foundation for ML models as stability and robustness in the manual annotation or monthly validation process is utilized as an input for model training or re-training.

The rest of this paper is organized around 5 sections. Following the introduction, section 2 identifies the research questions, dataset and methods used, and experiment design to test each question. Section 3 lists the results obtained from the numerous experiments. Section 4 discusses the impact on consumer price indices and comments on the processes that could be applied in production. The research concludes with ideas for further research that would support the prices indices field.

2. Experiment design

2.1. Research questions

To investigate misclassification on price indices, this paper focuses on answering four key research questions which encompass different aspects of the classification process.

Research question 1: How can human annotator consistency or inconsistency guide NSOs in designing labelling or validation processes?

In price indices, products need to be assigned to custom taxonomy categories, with category definitions that are sometimes heterogeneous or subjective and thus challenging to categorize consistently. Identifying the level of annotator agreement across a dataset as a whole helps NSOs establish the likely ceiling that ML model performance can reach (referred to as Bayes error rate, which is analogous to irreducible error), indicating a minimum level of misclassification that can reached with ML models without any quality control. Furthermore, understanding levels of annotator agreement for each category in the dataset supports NSOs in designing processes for effective allocation of resources for annotation and monthly validation. Specifically, multiple annotators can be assigned to label or validate categories known to need more robust approaches, whereas simpler categories may require less investment. Furthermore, better guidance and training material can be developed to create consistency between annotators. Considering the experience of the Office for National Statistics (ONS) (Greenhough, Martindale and Sands 2022), it is expected that annotator agreement will be high overall, however disagreement will be driven by specific categories. Finally, understanding the Bayes error rate for each category can help guide NSOs conducting ML model training, by better understand the trade-off of reducing the bias or variance errors of the ML models.

Research question 2: Can misclassification affect an elementary price index?

While methodological research has identified that misclassification could lead to measurement error in statistics such as counts and total turnover compiled after classification, effect on price indices has been understudied. To justify manual validation and guide any discussions on how to design resource-effective targeted quality control processes (De Waal and Scholtus 2011), misclassification needs to be demonstrated to be able to affect the elementary index—the building block of the CPI—both in one time-period and over time. Based on previous findings that demonstrated the impact of lower quality classifiers on price indices (Greenhough, Martindale and Sands, Modernising the measurement of clothing price indices using web-scraped data: classification and product grouping 2022), this research hypothesizes that misclassification affects the elementary price index.

Research question 3: Does performance of ML classifiers decline due to dataset drift?

Understanding the rate of data drift faced in ML classifiers applied on alternative data is key for NSOs as high-drift scenarios require more robust monitoring processes, more involved validation processes, as well as more robust

MLOps investment to develop an automated and routine model retraining process. This research expects to find moderate and gradual model decay rather than strong and very rapid changes, as the underlying generating function of the data which produces the product descriptions in alternative data and natural language is likely to shift slowly over time. Furthermore, the authors are not aware of a study on this topic within price indices literature, hence this experiment could begin a conversation on the development of MLOps best practices within the price indices field.

Research question 4: Which outlier detection methods are useful for NSOs to utilize to maintain classification performance?

If misclassification occurs and performance of classifiers declines over time, NSO should operating at-scale typically design 'selective editing' processes rather than validating each unique record (De Waal and Scholtus 2011). Thus, NSOs need to consider various outlier detection methods to flag targeted records for manual validation – such as price outliers, confidence outliers, or flagging a larger proportion of products in smaller categories. Basket weights could also be used to flag products that have proportionally high weights in the CPI; however, this was not investigated in this study. This research hypothesizes that flagging unique products based on low classifier confidence would be a promising recommended first step, both to identify potential misclassifications as well as to improve classifier performance, as confidence flagging is inherently aligned with margin-based active learning, a common way to increase the learning rate for ML models (Settles 2009).

2.2. Data and methods[1]

To answer these research questions, we utilize a web-scraped dataset collected from seven Clothing and recreational-goods retailers. While much of to focus for NSOs is on scanner data, we selected web-scraped data for a few reasons. Firstly, these retailers are available in a fully labelled form and are of moderate size, spanning a sufficiently long period. This allowed the research to evaluate the posed research questions and not incur too much compute cost or research time. Secondly, given the number of retailers present in this dataset allowed research to validate research questions across multiple retailers. Thirdly, as supervised ML models utilize the natural language in the data to predict which category each unique product belongs to, the task that is inherently similar between scanner and web-scraped data. At the same time, web-scraped data has some limitations and additional research would be required to validate results on other alternative data. For instance, product weight in scanner data is an important variable that could be included in testing research question 4 on outlier detection.

For the purposes of the research, we selected two subsets from within this dataset to support the research questions while also taking advantage of existing processes within the Canadian CPI to minimizing additional resource needs. The first subset consisted of a sample of 19,569 unique products, scraped from four retailers between June 2018 and December 2019. This dataset was labeled independently by three of four annotators to support research question 1.[2] All labelling for was done to the lowest level of CPICOM, the taxonomy utilized within the Canadian CPI.[3]

The second subset was used to support all other research questions. Subset 2 contains products and prices scrapped from retailers between June 2018 and December 2021. An initial dataset of 14,309 unique products

[1] All experiments were done in Python 3.9 with sklearn, pandas, numpy, and plotly for all calculations.
[2] This annotator experiment was also conducted as part of the initial labelling efforts for these retailers and trialed Active Learning to develop a cost-effective labelling process. Research on the topic is forthcoming.
[3] See the Canadian CPI reference paper for more details https://www150.statcan.gc.ca/n1/pub/62-553-x/62-553-x2023001-eng.htm.

were labelled by CPI Production experts, on which a Support Vector Machine (SVM) model was developed and deployed (Dongmo-Jiongo 2021). All remaining unique products not labelled were classified by the SVM model and manually validated. In production, Statistics Canada maintains full validation by price experts that are highly experienced and focus on their portfolio industry, thus this research will consider the validated labels as correct (ground truth) for the purposes of all research questions. While Statistics Canada continues to utilize this dataset in production and validate all new unique products received, this research will focus on three retailers for a 3.5-year subset of the data between June 2018 and December 2021. This subset contains a total of 155,254 unique products, broken down as 99,202 unique products up to and including December 2019, and 56,052 additional unique products and approximately 20 million price observations from January 2020 onwards. All price index and model experiments are carried out on the period between January 2020 to December 2021.

To support research question 2 on misclassification, we introduce various levels of misclassification on dataset subset 2 in both a random and simulated way. Random misclassification was introduced at various thresholds by altering the category of a selected number of unique products, with the number aligned to the threshold level. While random misclassification is a depictive initial test of the concept, realistic models do not misclassify randomly but instead tend to make mistakes where categories are heterogeneous or highly related to other categories in the dataset. Thus, we developed a simulated misclassification method whereby categories were rated according to how often a typical supervised ML model makes mistakes (see model architecture below). We apply misclassification at various thresholds at the total level, with misclassification allocated first to categories where mistakes occur more often and less on categories where ML models tend to perform well. This approach is based on assumptions that these model-error patterns scale to a certain extent with the performance of the model; an assumption that will hold to a lesser extent the further we depart from the actual model performance. Specifically, as performance declines, such as due to smaller training datasets, we assume that a classifier performs worse on categories that are challenging to differentiate between others but continues to perform well on categories that are simple to differentiate. At a certain point, the proportion of mistakes between challenging and simpler categories will no longer hold, thus the simulated methods should not be used beyond a moderate performance. Furthermore, as the proportion of mistakes between categories is dataset- and model-specific, this method is limited and is meant to be used to demonstrate the application of misclassification over time in a more realistic level than randomly.

To proxy a production scenario, we apply simulated misclassifications for all experiments over time, and ensure that a mistake persists once a product is misclassified as long as it is in the sample. This portrays a production setting where new products are classified and validated (using various methods as referenced in research question 4 on outlier methods), but then are not likely to be reviewed if the product is in sample. In this way, if less than 100% of new unique records are validated, the proportion of misclassified products can build up in the elementary aggregate category over time.

Research question 3 (model decay) and research question 4 (outlier detection methods to mitigate decay and support retraining), require a more realistic scenario, including the need to retrain realistic production models. As such the research needs to select a representative production model, for use in these experiments. We selected a Support Vector Machine (SVM) classifier, adopted word tokenization, custom stop word removal, and TF-IDF vectorization – an approach that typically is highly performant (Harms and Spinder 2019) and popular for many NSOs (Greenhough, Martindale and Sands 2022; Van Loon 2020; Hov 2021; UNECE 2021). Furthermore, this model architecture has also been utilized in production in the Canadian CPI for the Clothing index since January 2020 and has proven highly effective (Dongmo-Jiongo 2021). To perform the experiments required to answer the questions and allow for the model fitting and frequent re-fitting during re-training, we omit hyper parameter tuning and instead select a combination of fixed model hyper-parameters, known to be performant on this dataset. Variation

in classifiers is thus restricted to the training data used to fit the model. While the approach of not completing hyperparameter search each time is limiting, internal research has shown that hyperparameters of the model algorithm of consideration are not highly sensitive over time given the data considered and thus the exclusion of this step from experiments was not seen as harming the representativeness of the findings. When periodically refitting the classifier, new, reviewed products are added to the training dataset, creating an ever-expanding training dataset used for the duration of the experiment. While not the exclusive option NSOs face, this approach was selected as a default benchmark. Further research at each NSO is recommended to select the appropriate method in production.

To evaluate misclassification, we measure two separate dimensions. To evaluate the performance of ML models, we monitor the sample weighted F1 score of a model on any given month. F1 score is a harmonic mean of precision ($\frac{true\ positives}{true\ positives+false\ positives}$) and recall ($\frac{true\ positives}{true\ positives+false\ negatives}$) and is widely used by NSOs as an evaluation metric (Greenhough, Martindale and Sands 2022). Weighting is done through class-specific weights based on the class support (sample size). To provide a representative example of the impact of misclassification on price indices over time (research question 1), a GEKS-Jevons multilateral index is calculated at the elementary class level. We selected this method as a representative one for three reasons. Firstly, multilateral methods are widely used as even with an extension method, they are considered preferable to bilateral methods due to lower levels of bias introduced over time (Chessa 2021; Fox, Levell and O'Connell 2022). Secondly, they are also recommended for unweighted web-scraped data, as well as for clothing use cases (Greenhough, Martindale and Sands 2022) similar to the datasets used in this paper. Finally, as multilateral methods take into consideration the prices and, potentially, quantities of products between multiple periods, misclassification could affect multilateral indices more and at increasing levels over time (if misclassification increases over time and is not corrected) than bilateral indices, which instead focus only on matched set of products between the base period and a period in the future and are thus likely to remain more consistent (at least prior to chaining).

We simulate a production scenario by using a window-on-published splice as it often performs better and exhibits less bias than the classical window splice, a preference in existing literature. A 13-month window is chosen for two reasons. On the one hand it is well established in empirical research and its practicality; this choice allows seasonal products and pricing to be captured throughout an entire year. On the other hand, while larger window sizes are recommended, due to the 24-month time period available in dataset subset 2, a 13-month window with an extension method was a more representative application of a realistic production setting than selecting a 25-month window without extension. For comparison however, a 25-month GEKS-Jevons is also calculated to demonstrate the role of classification within a window. A discussion on the difference in outcome between different window lengths and extension methods is not included in the full analysis to separate out the discussion on misclassification detection and index parameter optimization strategies. Similarly, proxy weights were not utilized to separate out the experiment from a discussion on proxy weights within web-scraped data.

2.3. Experiments conducted

For each research question, a specific experiment or set of experiments were designed.

2.3.1. Cross-Annotator Agreement experiment

To support research question 1, a detailed annotator consistency experiment was designed. This experiment served two purposes: to test how annotators performed over different retailer datasets and set a benchmark, and to design a high-quality process for future annotation or validation within the program. Prior to beginning the annotation process, a detailed dictionary and guidance for human labellers was prepared, detailing definitions of codes as well as inclusions and exclusions. The annotation process involved two rounds, similar to a Delphi

technique for labeller consensus. First, each unique product was labelled independently by 3 initial annotators. If there was any disagreement between the three, a second round was used where a 4th annotator would be involved to arbitrate between the proposed labels (unlike the initial 3, the 4th annotator could see the proposed labels) or select another category. For products with no disagreement between the initial 3 annotators, the consistent label was considered the correct one; for products with any disagreement, the 4th annotator's decision was considered final as they could discuss and arbitrate the subjectivity and make recommendations where products should be placed in the future. For this role, a senior and highly experienced domain expert working on CPI Production was selected as the 4th annotator. Within the initial 3 annotators, 2 were similarly domain experts, whereas the last initial annotator was not experienced in the domain and relied quite heavily on the dictionary and guidance developed. Including only one less experienced annotator was done for two reasons. On the one hand as there was insufficient resources to engage a larger group, while on the other still supported an evaluation of how a larger body of less experienced annotators would likely perform, as well as support Statistics Canada in developing robust annotation guidance for the large volumes of alterative data still needing annotation.

2.3.2. Misclassification experiment

As the first step of calculating consumer price indices is to calculate an elementary price index, an experiment was designed to test misclassification at this level. Initially we simulate the impact of misclassification on a single elementary aggregate class, within a single reference period. First, for a specific retailer, we compile the set of all products observed in both the reference period, t_1, and the base period, t_0, denoted by S_p. We calculate the price relative for each product as the ratio of prices, p_1/p_0. The "true" index for an elementary aggregate class is taken as the geometric mean of the relatives, for all products assigned to that specific class with no misclassification.

To introduce misclassifications, we randomly re-assign classes to a selected number of products in S_p. For a single elementary aggregate class, misclassifications are introduced by either randomly removing products from the class (simulating false negatives leading to a decreased recall) or by randomly adding products from a different class (simulating false positives leading to a decreased precision). In each simulation, the number of products misclassified is selected to target a specified sample weighted F1 score for the elementary aggregate of interest. Once products are misclassified, we again calculate the index for the target elementary aggregate as the geometric mean of the relatives, for all products currently assigned to that specific class. This simulation is computed 1000 times each for a single elementary aggregate, at various levels of misclassification, to produce a simulated distribution for the calculated elementary index at each level of misclassification.

Secondly, we simulate the impact of misclassifications over the entire 24-month production period. Each month a defined fraction of the new products, first observed that month, is re-assigned a different class from their true class. In this experiment, misclassifications are introduced in a simulated, non-random way, i.e., the products selected for misclassification are based on the empirical accuracy of a production classifier, for that elementary aggregate. Additionally, the types of classification mistakes introduced are not random but based on the observed classification errors of a production classifier. This simulation is conducted three times to compare the calculated index using different methods at varying levels of simulated misclassification.

2.3.3. Model decay and retraining experiment

To evaluate decay in model performance from data drift, three scenarios are evaluated. Firstly, to assess whether decay occurs due to data drift, we train and deploy a model based on labeled data between June 2018 and December 2019. The trained model was then used to classify all new unique products that enter the sample every month for two years, with no validation or retraining. As the ground truth labels are known, over the two-year period, various classification metrics were calculated each month, using the predicted and true labels, including the sample weighted F1 score. While product turnover in clothing is known to be high, many products persist in

the market for many months and even years, thus some products were still correctly classified (from the December 2019 period or previously) in sample each month, hence we expect the overall F1 score to decline gradually. This scenario can act as a benchmark for NSOs of what happens over 2 years if no validation is performed and a model is not retrained.

Secondly, to simulate model retraining, a process that NSOs could apply in order to restore model performance to expected levels, we expand the first scenario but utilize the full ground truth labeled dataset as retraining data for various time periods, such as every 1, 3, or 6 months. For all cases, we add the full validated dataset into the original training dataset (thus expanding the dataset) and retrain the model. This scenario represents a second benchmark, a case where the ML model is often retrained but all new data is still validated. Any price index calculated from this dataset represents the "true" price index, as 100% of the products have been reviewed and thus assumed to be correctly classified.

2.3.4. Outlier detection experiment to mitigate misclassification

To mitigate reducible misclassification from model decay, we investigate suitable outlier detection methods that could be used to re-train models, thus helping develop a targeted quality assurance process after classification. We attempt four different methods of outlier detection to simulate a review process employed to detect misclassified products, each with a threshold parameter. In accordance with De Waal and Scholtus' selective editing principle (T. De Waal 2013), the goal of these selected methods is to focus attention on a so-called "critical stream" of records (i.e.: products most likely to contain errors) and identify misclassified products to mitigate the impact on the CPI and related publications. Methods are trialed individually and also compared together to provide a representative demonstration of their combination, however a detailed assessment of all outlier methods to select the most optimal set is not in scope, as the authors felt that this warranted a separate and detailed study. As each outlier detection method is evaluated to support model re-training, we adopt a 3-month retraining window as trialed in the model decay and retraining experiment (above).

The first of our methods is a simple random sampling flagging method whereby a fraction of new products are randomly selected each month for review. We alter the fraction parameter to randomly select 0%, 5%, 10%, 20%, 30%, 50%, 70% and 100% of all new products for the month. This method serves as a base to compare other methods; for instance, a method that flags about 30 percent of products in a targeted way should identify more misclassified products than the equivalent random flagging. A second use of random flagging is to create a sample for model evaluation which has low selection (sample) bias compared to other flagging methods due to random sampling property and allows for an effective evaluation of classification performance. Finally, the low selection bias of the random sample motivates its use in the retraining process, to avoid biasing the classifier model.

A confidence-based approach using model probability scores is also considered. Here, probabilities are derived from the classifier's distance metric via Platt scaling (built into the scikit-learn Python package). Classifier margin is calculated as the difference between the probabilities for the first and second most likely classes; lower classifier margin indicate that the classifier cannot effectively separate the top two classes. All products below a certain threshold are flagged and a higher threshold will flag more products. We test thresholds of 1%, 2%, 4%, 6%, 8%, 10%, 15%, 20% and 30%. Note that the classifier margin depends on the internal representation of the decision boundaries of the classifier and as such the flagging method results are classifier dependent. Furthermore, confidence-based methods could be considered an Active Learning method, selecting a targeted sample for use in the model retraining process (Settles 2009).

The third method we propose leverages counts. Counts was chosen as the price distribution of product classes with less products will tend to be more sensitive to misclassification, compared to those with more products. A

single misclassified product would have a higher relative impact on a class with a small number of observed prices, compared to a class with many observed products. Furthermore, categories with few products can easily be reviewed by NSOs, as low investment is needed to validate a small category. A threshold of 10, for example, means that if there are less than 10 new products observed that period, all new products in that category will be flagged. Count thresholds of 3, 5, 10, and 15 are used to flag in this instance.

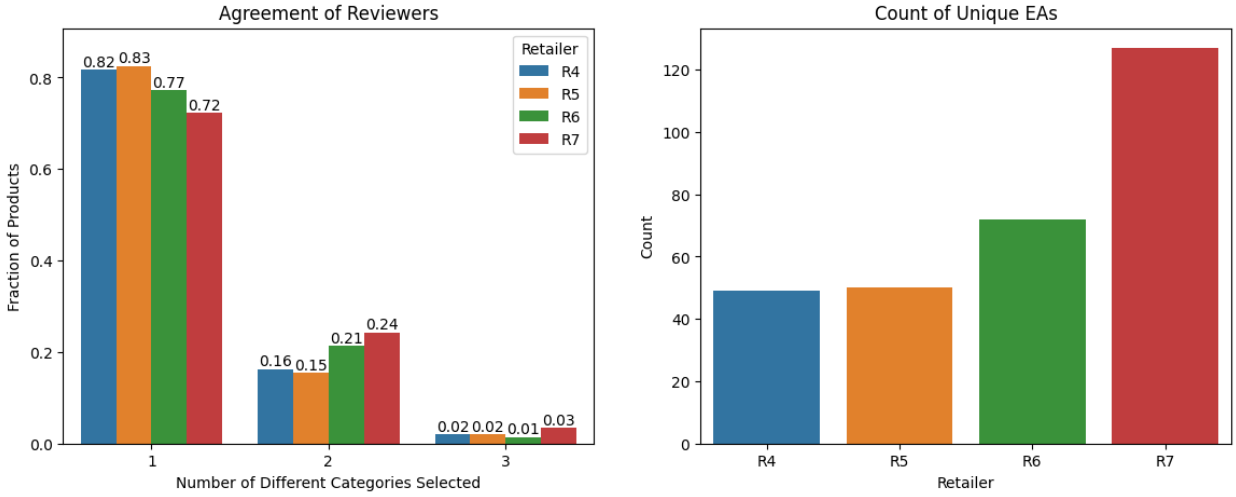
Our final method flags outliers based on the distribution of product prices within each elementary aggregate class. This flagging method uses a simplified approach, which considers prices of products being classified each period (new products). Once all new products are classified, the price distribution is calculated for each elementary aggregate, using the observed prices for all new products assigned to that class. The price outlier method then flags products from each class with prices that are above or below the specified percentile threshold; thresholds tested in this experiment were 5 and 10 percentiles. While this method utilizes a simple approach and has limitations in a fuller application in a production setting, it was included in the research for two reasons. Firstly, the size of the churn in the data, with an average 3,000 unique new products entering the sample each month, meant that simple setup of the experiment still allowed percentiles to flag a considerable number of products in practice. Secondly, as price outlier detection is an important area of focus for NSOs and is often used in production, any representative demonstration of various outlier methods should also include a price flagging method.

3. Results

3.1. Findings from cross-annotator agreement experiment

Our findings on annotator agreement are similar to the experience of ONS (Greenhough, Martindale and Sands 2022). Evaluating cross-annotator agreement (Figure 1), 78% of the time the initial 3 annotators agreed on the same category, 20% of the time two of the three agreed, and 2.3% of the time, each annotator chose a different category. This differed slightly between retailers, with retailers that were larger (offering a larger and less consistent range of products) and needed to be labelled to a larger number of categories (Elementary Aggregate category or EA), was associated with less agreement between annotators. Overall, the experiment agreement that was non-random as the experiment had a Fleiss Kappa of 0.84, a metric that compares raw agreement counts against levels that could be obtained by arbitrary labelling.

Figure 1: Annotator agreement by retailer, compared to number of categories labelled in retailer dataset



Similar to the ONS, this research found that consistency varied among categories, with some categories being relatively simple to label and had very high consistency levels, whereas other categories were very inconsistently labelled. Furthermore, similar to the ONS, there was no major correlation between annotator consistency and the number of unique products per category. Categories that were most problematic were often categories which are hard to define such as catch-all categories ("other footwear" labelled consistently only 8.3% of the time or "other children's clothing" labelled consistently 6.7% of the time), heterogenous categories which are hard to separate from other categories. Examples of such heterogenous categories are "children's winter outerwear" which was consistently labelled 34% of the time, and "Children's winter boots" which was consistently labelled 29.8% of the time, and "Women's casual pants and shorts" which was often confused with "Women's dress pants". At the same time, categories that were simpler to define, more homogeneous, and quite distinct from other categories were usually labelled very consistently. For example, "Men's sunglasses" was consistently labelled 100% of the time, "Children's shorts" was consistently labelled 98% of the time, and "Women's skirts" was consistently labelled 97.5% of the time.

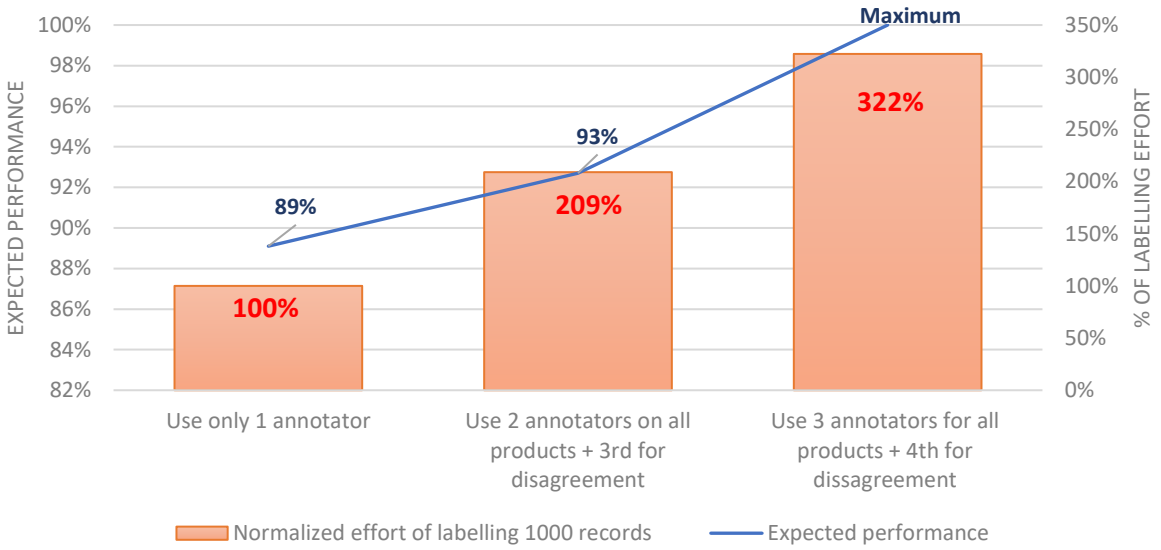
Evaluating more closely the possible reasons for this disagreement – specifically to see whether expertise affected an individual's performance and whether differences in expertise could be mitigated by better training or guidance – each of the 3 initial annotators performance was evaluated, and their choice of labels was compared to the final accepted label. The 2 experts with more contextual knowledge of the clothing domain showed an F1 score of 0.902 and 0.918, thus they showed a very high likelihood of selecting the final label considered correctly. In contrast, the less experienced initial annotator showed only an F1 of 0.845. At the same time, the less experienced annotator still was correct for most homogenous categories but tended to be less likely to select the correct category for catch-all categories or heterogeneous categories closely related to other categories.

A final phase of the experiment introduced a fourth annotator to validate the situation and attain consensus in order to reach a final decision on what category each product should belong in, as well as evaluate how a consensus process could be developed. 61% of the time, the fourth annotator chose a category which two of the three annotators also selected, 32% of the time the annotator decided for a label which one of the three annotators identified. Finally, 7% of the time the 4th annotator overruled all three annotators to select a previously unselected final label. As the labelling was done in batches, at the end of each batch annotators met to discuss this situation and confirm this decision, with the 4th annotator marking this as the final decision.

This finding leads to two takeaways. Firstly, redefining categories to be more homogeneous, minimizing the use of catch-all categories to the extent possible, and expanding the documentation and exclusions/inclusions dictionaries provided to annotators could mitigate some of this subjectivity. At the same time, fully eliminating subjectivity is not possible, a scenario that needs to be balanced with the need to define cost-effective strategies for both initial annotation and monthly validation. Thus, secondly, NSOs could explore developing a process where more complicated categories, after they are initially identified such as with a targeted sample as in the ONS experiment (Greenhough, Martindale and Sands 2022), are allocated more resources to attain high quality and consensus. We demonstrate this using our use case and the whole retailer dataset (Figure 2), normalizing the dataset size to 100% for comparison purposes. In other words, if all the records in the category (or the whole dataset in our case) is labelled, 100% effort is expended, whereas if each record is labelled by two individuals, 200% effort is expended. From our case study, if one annotator labels every record they are provided, they are expected to have an accuracy of 89.1% after labelling 100% of the dataset. Introducing a second annotator would require a doubling of the annotation effort, and the effort of both could help identify a subset of products that could go to a third annotator who would be responsible for finalizing the decision and reaching consensus. In our use case, we had 9% on average disagreement between two annotators, meaning that 209% effort would be necessary, however the process was expected to lead to 92.7% accuracy. Expanding this to having three initial

annotators for the dataset, with a fourth annotator brought in to reach consensus led to an investment requirement of 322%. While we stopped with four annotators for this research due to resource limitations, we consider the consensus process and the experience of the fourth annotator, while not perfect as to lead to 100% performance in practice, to have led to a conceptual maximum accuracy. An important consideration of this takeaway was that maximum performance may change over time as NSOs develop more homogeneous and more objective categories or improve their training material for annotators.

Figure 2: Expected accuracy of annotated dataset versus annotation effort required



3.2. Findings from misclassification experiment

This experiment shows that misclassification could affect the elementary (aggregate) price index, as misclassified products in an EA could shift the distribution of the price relatives of the whole EA. While this experiment was conducted on each retailer for multiple elementary aggregates, in multiple reference periods; the results presented below are limited to two representative elementary aggregates for retailer 2.

The following four figures (Figures 3 - 6) show distribution plots, highlighting the results of individual experiments. The true index for the selected elementary aggregate, as defined in section 2.3.2, is shown as a dashed vertical line. Each of the lines in the top plot, represent the frequency of observed values for the calculated index, based on the misclassification simulations; smoothed using kernel density estimation. In the sub plot underneath, each vertical line represents an individual observed value for the calculated index, from a single iteration of the misclassification experiment. Each figure represents a single retailer, a single elementary aggregate, and single reference period combination; with each line representing different degrees of misclassification.

Figures 3 and 4 both show how two different EAs that were respectively either above or below the average movement of the retailer could be affected. In both instances, the bias increases with decreasing F1; directionally the bias is towards the mean relative of all products within that retailer and reference period. Furthermore, the experiment shows qualitatively that both the difference between the mean of the distribution and the true index, and the standard deviation of the distribution, increase with the number of misclassified products. The mean shift is directionally towards the mean of all price relatives of all products in the specific reference period. In both cases, even a relatively high F1 score that NSOs could expect for models used in production could affect the EA by a few points. It should be noted that the quantitative change in mean and variance depends on the specific reference

period, retailer and elementary aggregate used in the simulation; these figures are intended as representative examples.

Figure 3: Random misclassification for an elementary aggregate class above the average movement of the retailer

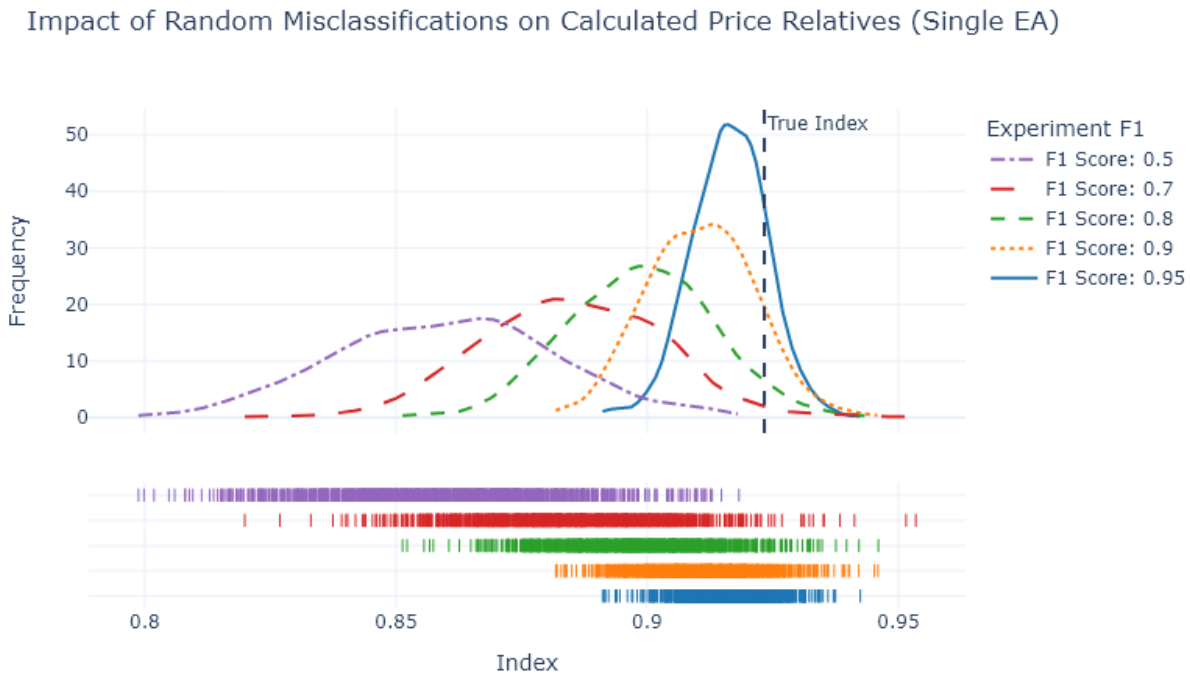
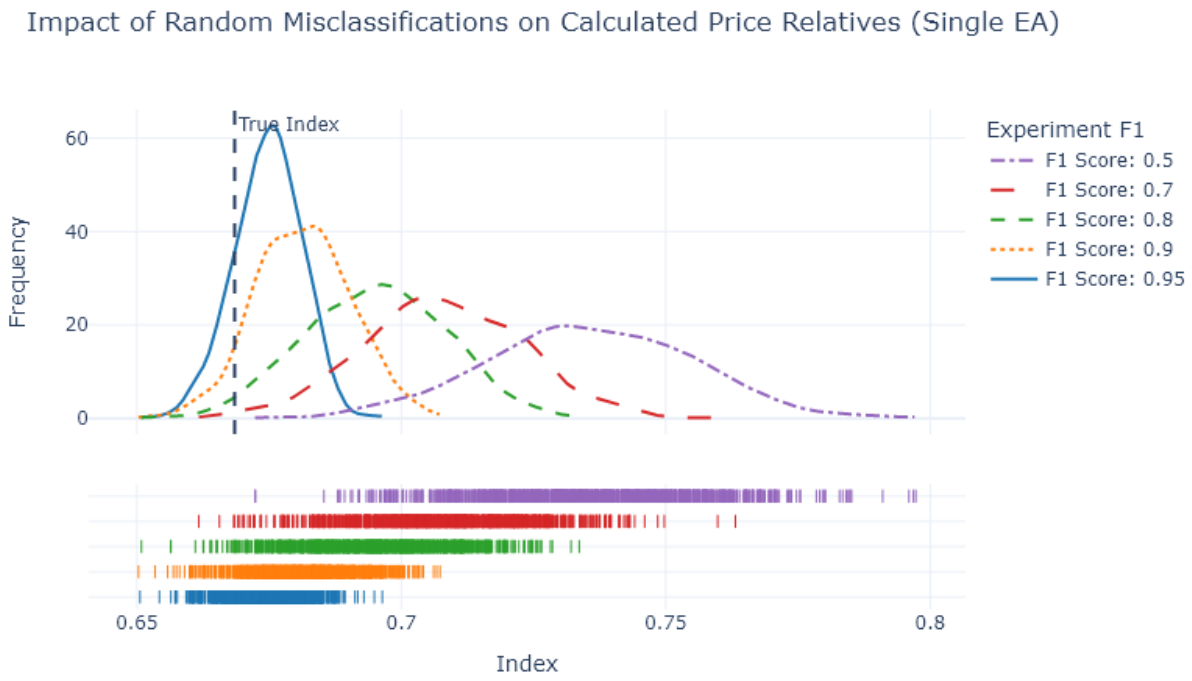


Figure 4: Random misclassification for an elementary aggregate class below the average movement of the retailer



While Figures 3 and 4 demonstrate the effect with an equal number of false positives and false negatives to achieve the stated F1, in practice misclassification is not balanced. A related experiment was thus completed to evaluate whether the type of misclassification influences the elementary index. Specifically, the same EA is chosen as in Figure 3, but precision is fixed at 1.0, and recall is decreased by randomly removing products from the class (Figure 5). Qualitatively there is less, or no mean shift observed, though the standard deviation of the distribution increases as the recall is lowered, signaling no bias but largely increased variance. Conversely, if the recall is fixed at 1.0 and the precision adjusted by randomly adding products to the class, we observe that the mean appears to be shifting lower as precision is lowered (Figure 6) signalling both bias and small variance.

Figure 5: Various levels of misclassification when precision is fixed at 1

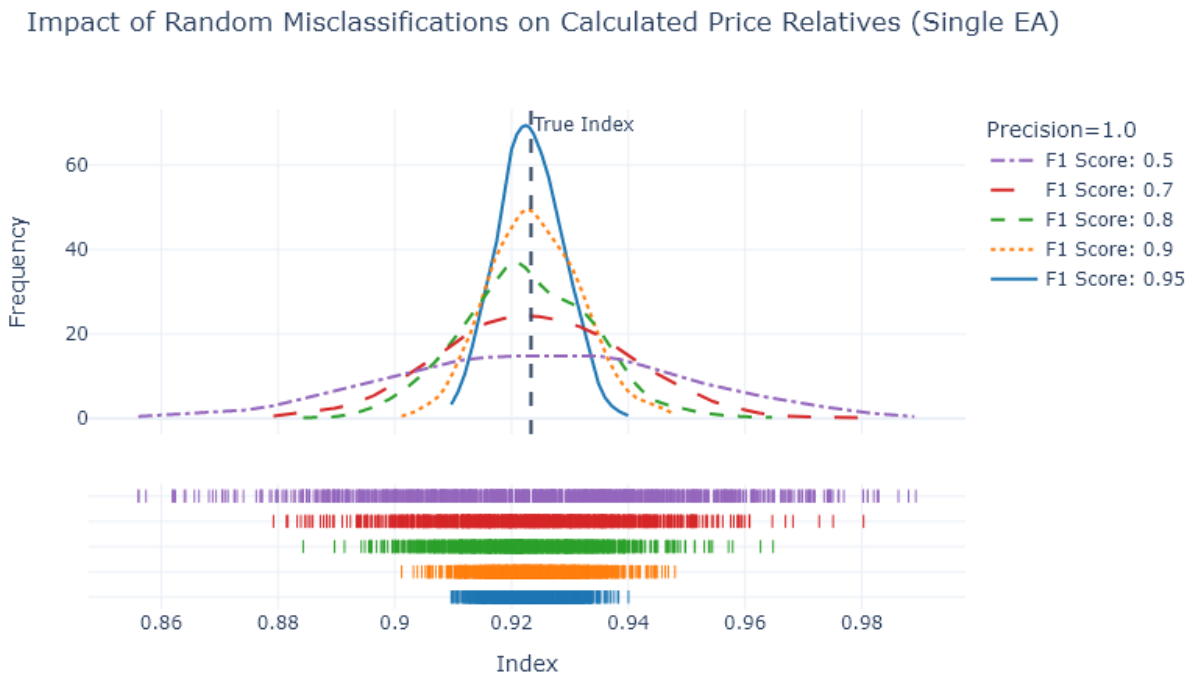


Figure 6: Various levels of misclassification when recall is fixed at 1

The experiment shows that that maximizing class precision is most important for eliminating bias in the calculated index; recall is important for reducing variance. Furthermore, a trade-off must be made between reducing bias and variance that may differ based on the nature of the category and quality assurance constraints. This is of course complicated by the nature of the multi-class problem; a misclassified product impacts two classes, the true class that it is removed from and the false class that it is assigned to.

To simulate the impact of misclassifications in a production setting, product misclassifications were introduced to new products observed in the 24-month production period of dataset subset 2. Simulations were introduced in a biased way, i.e., the types of classification mistakes introduced were based on the observed classification errors of a production classifier. The fraction of products misclassified was increased or decreased to reflect a high, moderate, and low performance of a production classifier over time. A category was chosen that had high nominal misclassification with another category that had at times different movement to visualize a representative situation of potential concern that NSOs would look to mitigate. Approximately 15, 50 and 95 percent of new products were misclassified in the high, moderate, and low performance scenarios respectively, for the elementary aggregate shown below. Figure 7 shows the cumulative F1 of the category over time as new products enter and leave the sample, and Figures 8 and 9 show a GEKS-Jevons index on this elementary aggregate, with a 13-month window extended over the remaining months, and a 25-month window, respectively.

Figure 7: Scenario of various levels of misclassification entering the sample over time to support Figures 9 and 10

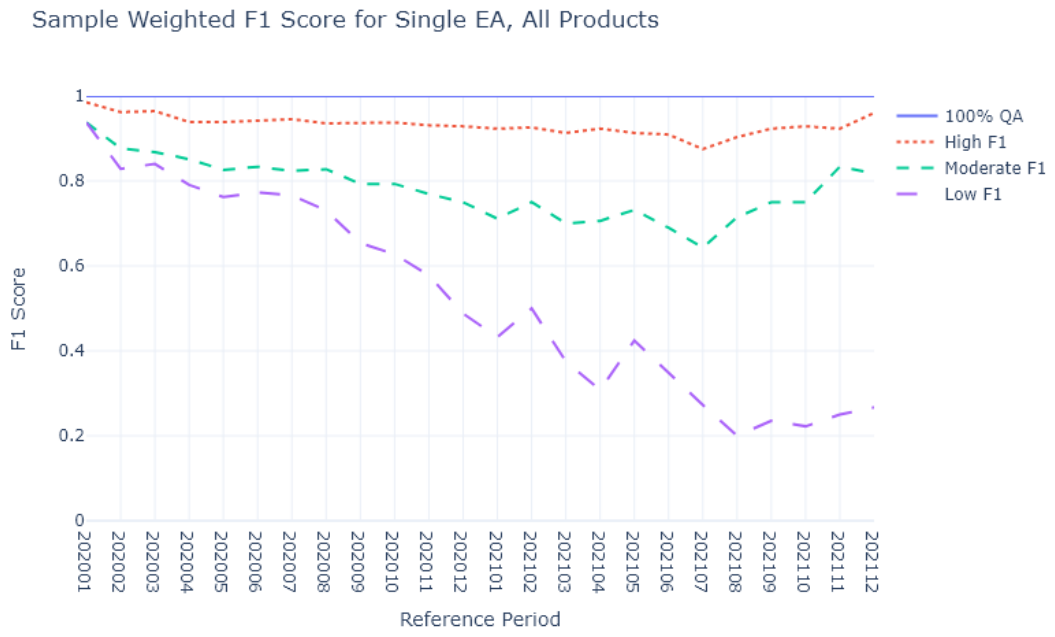


Figure 8: GEKS-Jevons with a 13-month window on various levels of misclassification over time

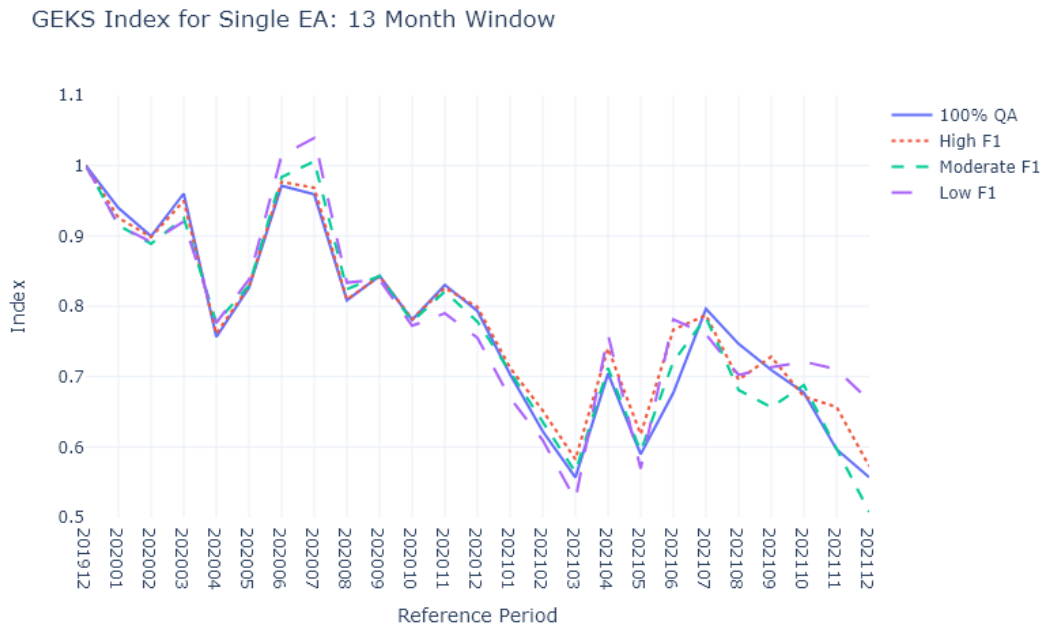
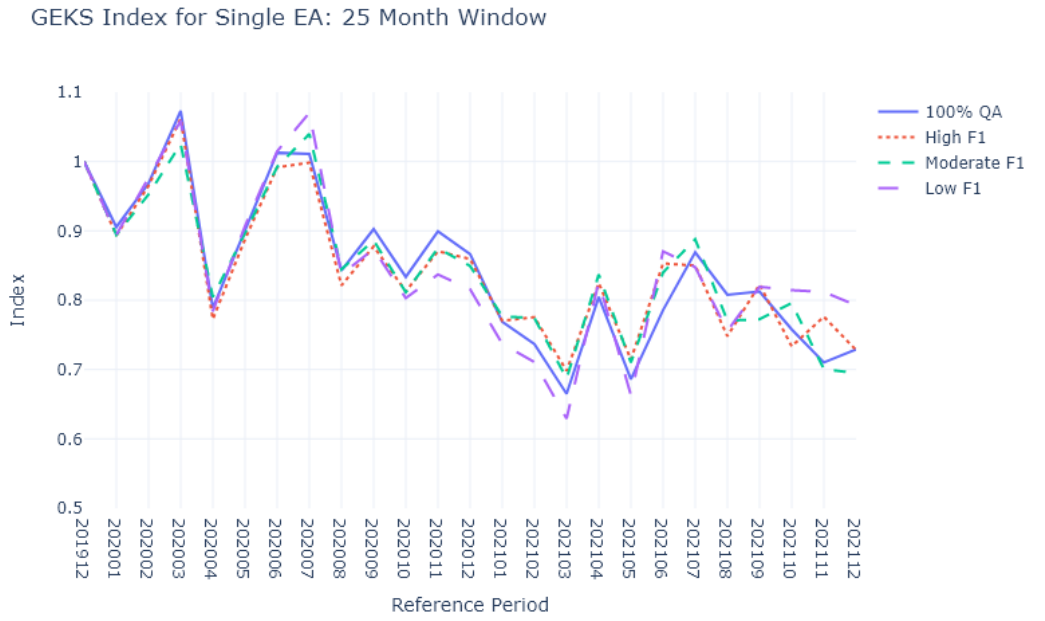


Figure 9: GEKS-Jevons with a 25-month window on various levels of misclassification over time



Our tentative results show that neither GEKS-Jevons index is highly sensitive to misclassification, however moderate and high levels of misclassification could still cause a significant effect over a moderate time period. Considering that the proportion of misclassified products could build up over time in a category, this situation underlies the importance of validation to check that misclassified records are caught. As misclassification could impact the index only if a proportion of wrongly classified products show a movement divergent from the correct products in that category, understanding the movement of categories the classifier typically confuses is key.

Further research is needed to demonstrate the impact on a wider set of cases and retailer datasets. Furthermore, a longer time period should be investigated as misclassification could build over time with different extension methods.

3.3. Findings from model decay and retraining experiment

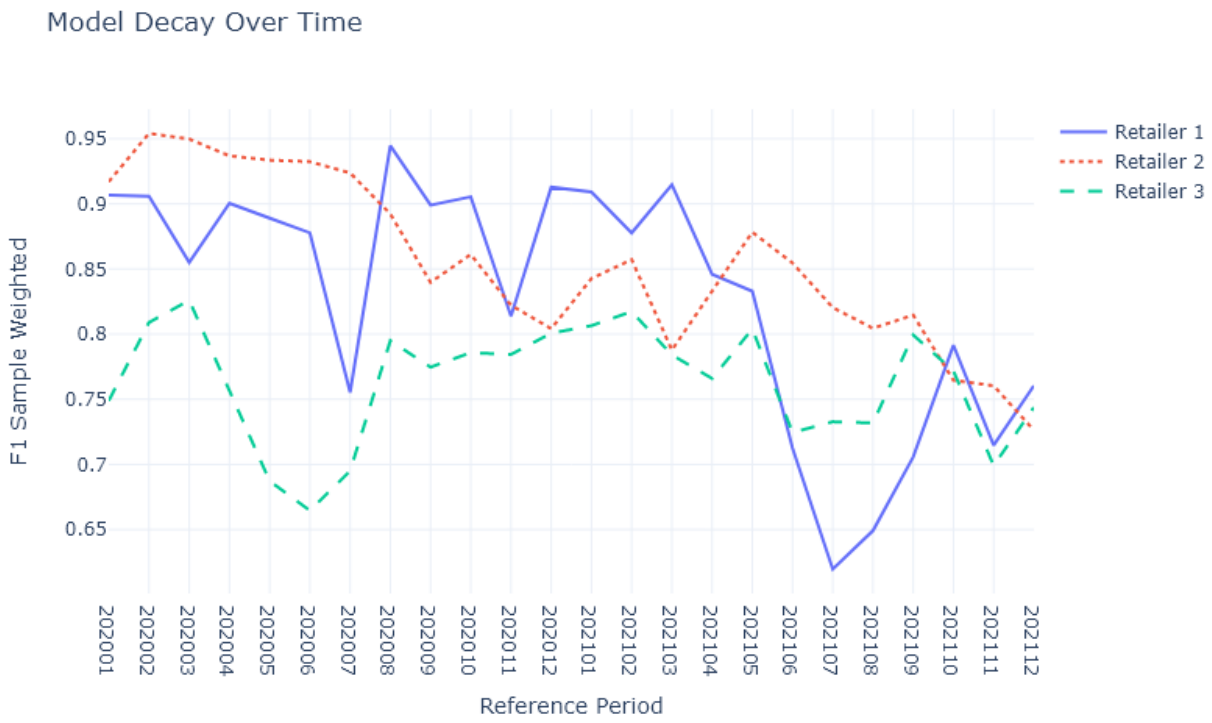
We train models with data up to 2019-12 and analyze the performance change using ground truth data over a two-year period. We compare performance decay of models for new products, with models trained on each of the 3 retailers under consideration (see Figures 10 and 11). We categorize the decays based on their characteristics using the following common principles (see Bayram, Ahmed and Kassler 2022 and references within): Probabilistic source of change, transition of change and severity.

First, we aim to categorize probabilisic sources of changes qualitatively and based on the understanding of the data generation process. We describe each product classified as an instance defined through its feature (covariate) vector X and its target (response) variable y. The feature vector encodes the product description while the target variable is the product class. Then the product distribution can be described via a probabilistic definition as the joint distribution P(X,y). Following the common notation and Bayram 2022 we define the posterior probability distribution as P(y|X). A classifier aims to learn this concept of mapping X to y, and any change to this relation or concept in new product data is called concept drift. This drift invalidates the learned concepts leading to performance decay. The learned concept of product description mapping to classes might be fairly robust to change over time, excluding planned changes to the class hierarchy. Note however we expect that new products with new descriptions change P(y|X) (perhaps in a P(X,y) sub-region) and the learned concept of the classifier becomes outdated. Additionally, we expect a change in P(X), defined as covariance drift, as the distribution of product descriptions in each period vary due to assorted reasons such as seasonality, and new products entering the market. We also observed a change in the probability distribution of the classes P(y), defined as prior probability, due to similar reasons as stated above, leading to significantly different class distributions between reference periods. In some cases, class counts reduce to a few counts per class, or even zero counts. These different probabilistic sources of change and their complex interplay result in the various observed patterns of model performance changes and transition patterns for the 3 retailers we are investigating.

We observe drift transition patterns which differ between retailers and over the 2-year time period. For example for the entire 2 years we find retailer 2 to show a clear gradual decreasing performance trend, while retailer 3 does not appear to show any decreasing trend. However, for all retailers, gradual drift occurs for at least a few periods. We also find sudden drifts and drop in performance, e.g. for retailer 1 on July 2020 and November 2020. We attribute these different transition patterns to the diverse changes to product offerings. The nature of model decay may be correlated to the type of products sold at a particular retailer; this phenomenon would need to be studied in more detail and on more retailers to draw specific conclusions.

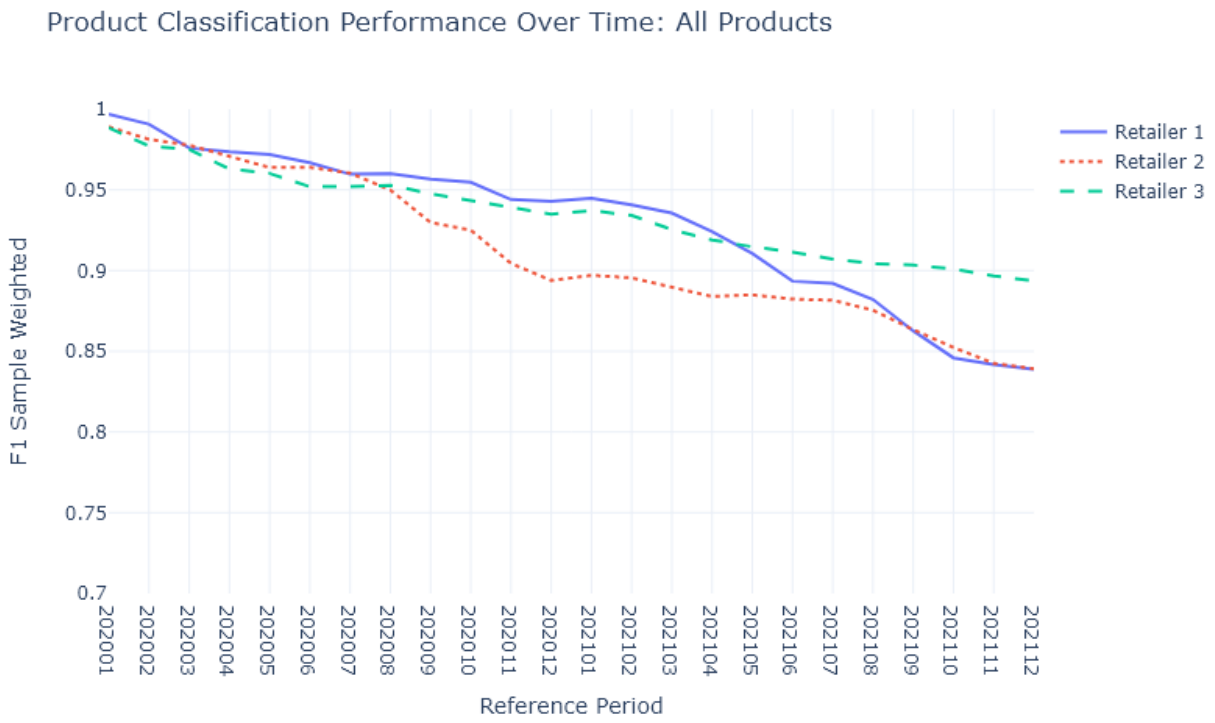
To judge the severity of the drift, we analyze the sample-weighted F1 score and find high variance across periods with changes of up to approximately 10%. Additionally, gradual, and strong drift over multiple periods result in drops in performance, for example of more than 25% for retailer 1 in 2021. We attribute this to the diversity of new products appearing due to unknown context (Widmer and Kubat 1996). Those strong performance drops and the sudden drift patterns justify re-training models frequently. Note that as our class assignment of existing products are not likely changing, severe concept drift (defined as class change of all products will not occur, and therefore not impact CPI calculation (Figures 8 and 9).

Figure 10: Classifier Model Decay Over Time



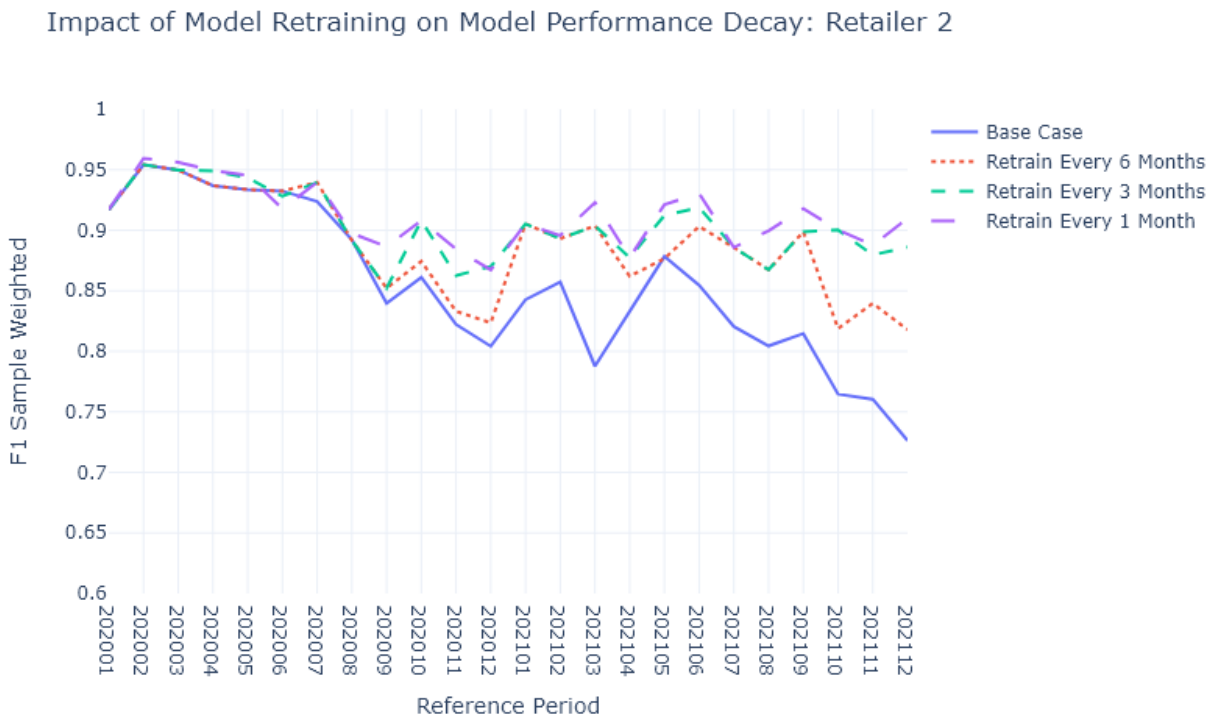
The decreasing performance on the new products is affecting the performance of the whole sample with all products of the reference month. The compounding effects of increasing number of misclassifications from month-to-month results in a relatively smooth decrease of the total performance, as shown in Figure 12. The 100% validated data from the initial time period is increasingly being diluted by the wrongly classified new products. This resulting error of all products directly impacts the CPI. This emphasizes the potential effect in the absence of a quality control process, including retraining. Note that while only sample weighted F1 is presented, we find the same trend in the dataset, with respect to both precision and recall.

Figure 11: F1 Score for All Products Observed in Each Reference Period Over 2-Year Period



To address the observed model performance decay and mitigate the impact of low model performance on the elementary prices index, the model can be periodically retrained. Figure 12 shows the classification performance on new products where the model was periodically re-fit with new data. We assume 100% product validation, meaning retraining occurs with all new products up to the month of retraining. The base case corresponds to no retraining, as shown above. One can observe that with periodic refitting, the classification performance can be stabilized over the 24-month period. More frequent retraining shows to be beneficial with less benefit on the time horizon between one and three months. This justifies a balanceing of the costs of retraining with the improved performance over that horizon span. We find similar behavior of performance improvements for retailers 1 and retailer 3 (similar to Figure 12) indicating a similar underlaying data and drift generating process.

Figure 12: Impact of Model Retraining Frequency

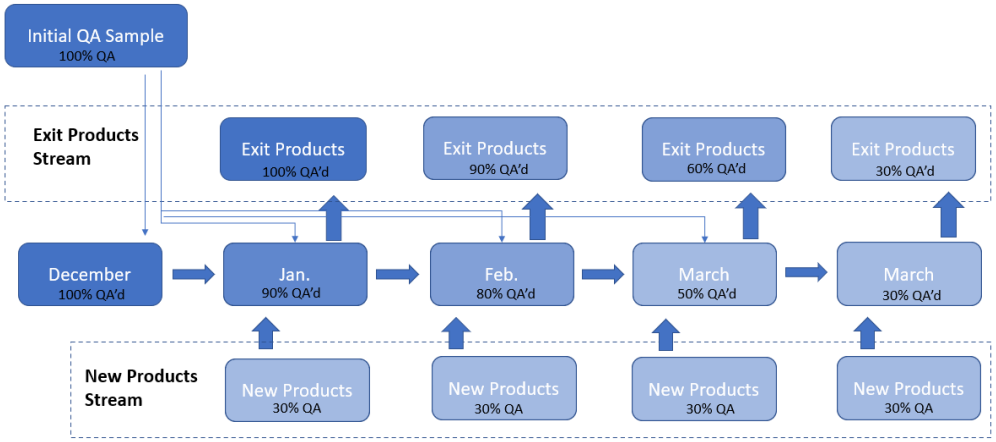


3.4. Findings of outlier detection experiment

In this section we analyze the effect of different flagging methods on performance based on data from retailer 2. We find similar qualitative results for retailer 1. Due to compute limitations, only select experiments were performed on retailer 3, though results were also consistent.

In the absence of 100 percent Quality Assurance (QA), as new products enter the sample and old products exit, an increasing level of misclassifications will enter the sample. Over time, as products from the initial 100% QA period (prior to 2020-01 in this experiment) exit the sample, the classification accuracy (and F1) will approach that of the classifier. Flagging and reviewing new products via QA can assist in reducing the number of errors that accumulate each month in the sample. The conceptual trend from 100% QA to a final state with constant QA fraction, defined through the classifier's performance, and a fixed 30% QA rate, as a demonstration, for new products is shown below (Figure 13).

Figure 13: Conceptual overview of how 100% proportion QA data would transition to the fixed QA rate over time, with a decreasing QA'd fraction in exit products and a constant QA rate for new products.



The introduction of an arbitrary flagging method will catch a percentage of misclassifications $\epsilon_{flagged}$ for correction. If we assume human annotators to be 100% accurate at correcting flagged products (assuming we design a process based on human performance to catch all mistakes), the new classification accuracy will approach the accuracy of the classifier A_{model} plus the fraction of misclassifications flagged, given as follows:

$$A_{errors\ flagged}\ (new\ Products) = A_{model} + \epsilon_{flagged} \times (1 - A_{model})$$

The accuracy on all products $A_{errors\ flagged}(all\ Products, t)$ for period t will depend on $A_{errors\ flagged}$ as follows, given $F_{new}(t)$ as the time-dependent fraction of new products, which will approach 100%:

$$A_{errors\ flagged}(all\ Products, t) = F_{new}(t) \times A_{errors\ flagged}(new\ Products) + \big(1 - F_{new}(t)\big) \times 100\%$$

The perfect flagging method would therefore flag all misclassified products ($\epsilon_{flagged}$ = 1). The most efficient flagging method would allow to flag *only* misclassified products to minimize the QA effort.

In the following we simulate the application of random and uncertainty flagging through 24 periods which result in different percentage of misclassifications $\epsilon_{flagged}$. We analyze the performance on new products and all products. Each period, a designated fraction of the new products was flagged for review, which automatically assigned the correct class to them regardless of the classifier prediction. This assignment simulates review by a human, in a production setting. Every three months, the model was re-fit using the original training data, plus all flagged products up to that date to simulate a production retraining process using validated data.

3.4.1. Random Flagging

We sample a fixed percentage of products F_{rand} from all new products in each period, which allows to flag an equal percentage of errors, leading to

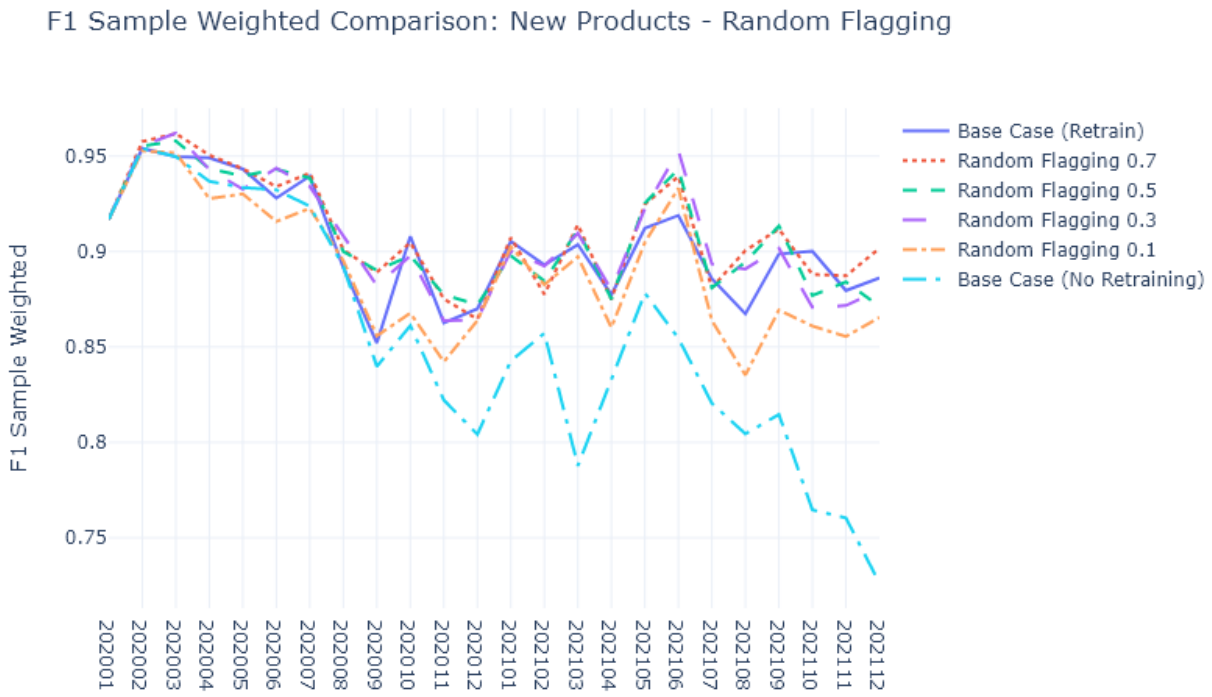
$$A_{random}(new\ Products) = A_{model} + F_{rand} \times (1 - A_{model}).$$

For instance, flagging 50 percent of total products should flag approximately 50 percent of misclassified products.

To track the performance of the classifier on new products and the expected performance decay, we analyze and present its performance *prior to flagging*. Model performance with respect to the amount of data randomly flagged is compared (Figure 14). The two bounding base cases represent the cases with no flagging nor retraining

(strongest decay) and the case where 100% of new products are flagged, reviewed, and used to re-fit the model every three months (base case with retraining).

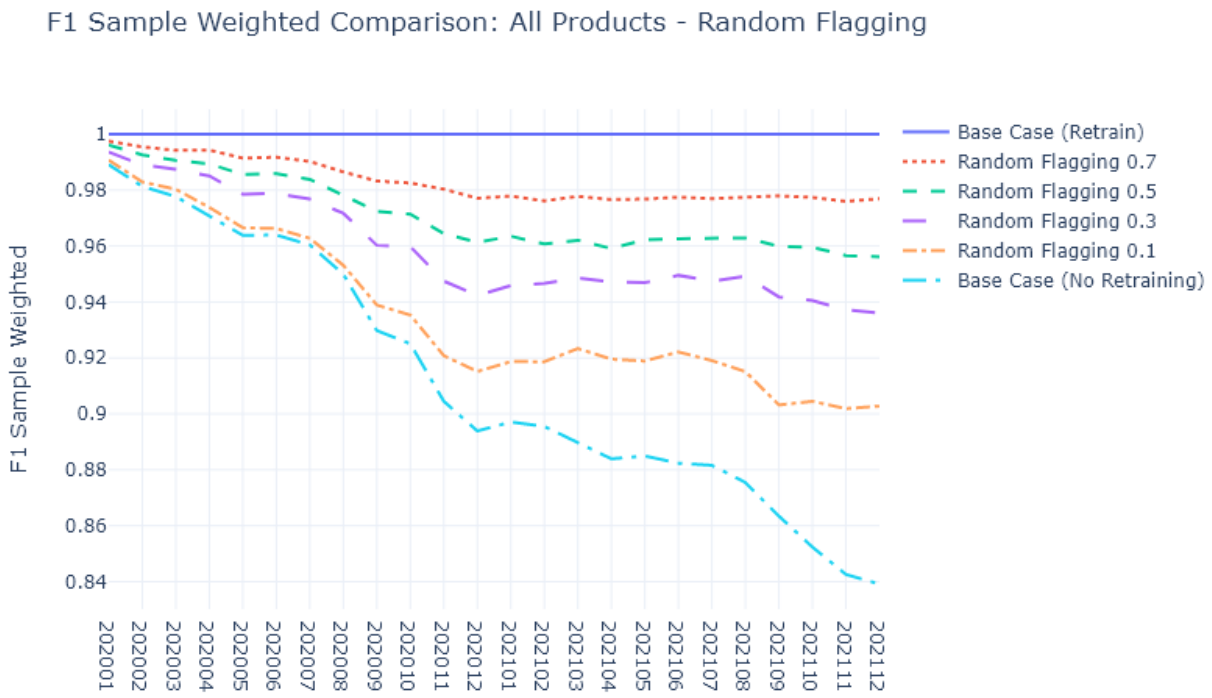
Figure 14: Impact of Random Flagging on Classifier Performance (with Refitting) for retailer 2. Shown is the model performance without considering the performance improvement from outlier flagging.



Flagging products each month provides additional and more recent training data for refitting the model and consequently the model performance improves, similar to the findings in Section 3.3. We find that more random flagging increases the classifier performance. Additionally, just randomly flagging 10% of new products for retraining strongly improves the classifier prediction performance over time, preventing the strong decay observed in case of the base case without retraining. Further flagging of 50% of new products or above has mostly very small impact on the overall performance.

While low levels of random flagging are sufficient to keep performance high on new products, classifier errors still accumulate over time: initial quality assured products leave the all-products sample and are replaced by the new products with classification errors, as discussed above. We analyze this accumulating effect as it impacts the performance of the whole sample of all monthly products in Figure 15, with new products corrected *only through* QA to demonstrate what happens with the overall cumulative F1 with random flagging without model re-training. As shown in the monthly F1 plot (Figure 14), the base case (no retraining) in Figure 15 for cumulative products shows strong continuous decline in performance towards the classifier accuracy. Thus, random flagging will both improve the classifier performance as well as reduce the number of misclassifications, leading to a much better F1 score for all products. The new QA'd products and the exiting products from the initial 100% QA'd data (Figure 15) together lead to a smooth, less pronounced decrease compared to the performance on new products only (Figure 14). Additionally, we observe an increasing overall performance for larger fractions of random sampling which is, in relative terms, is much more pronounced than the gains observed for new products only (Figure 14, noting the scale difference).

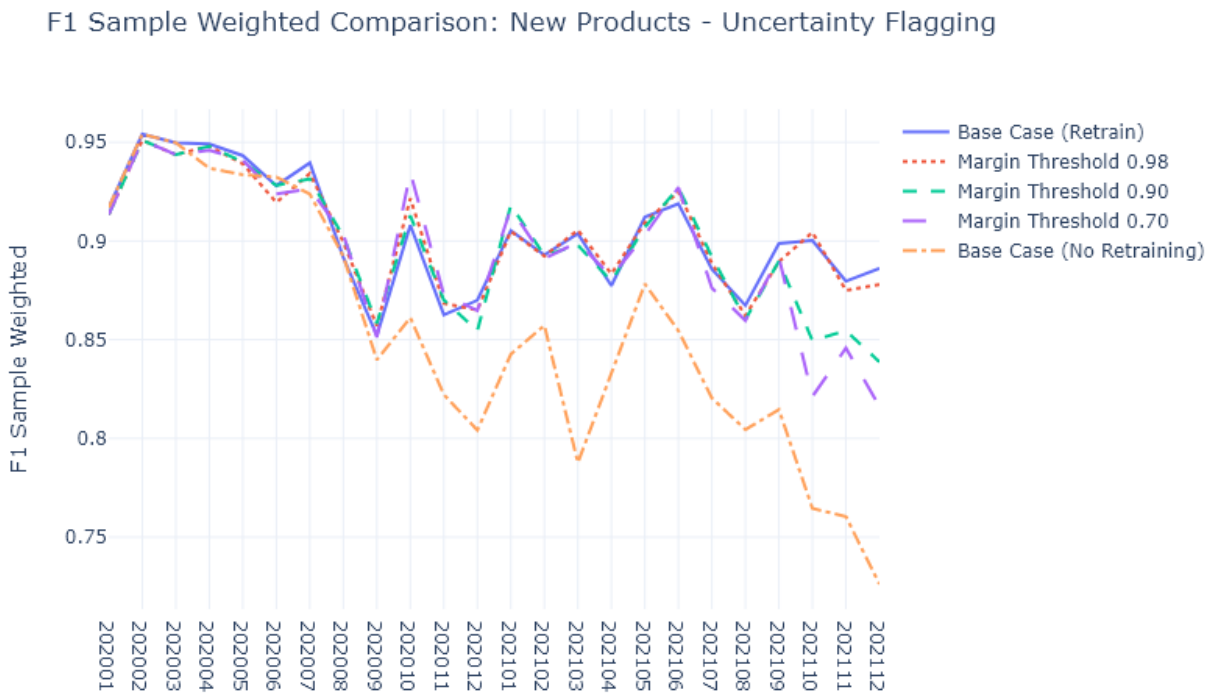
Figure 15: Comparison of Random Flagging Percentages on F1 Score for All Observed Products



3.4.2. Uncertainty Flagging

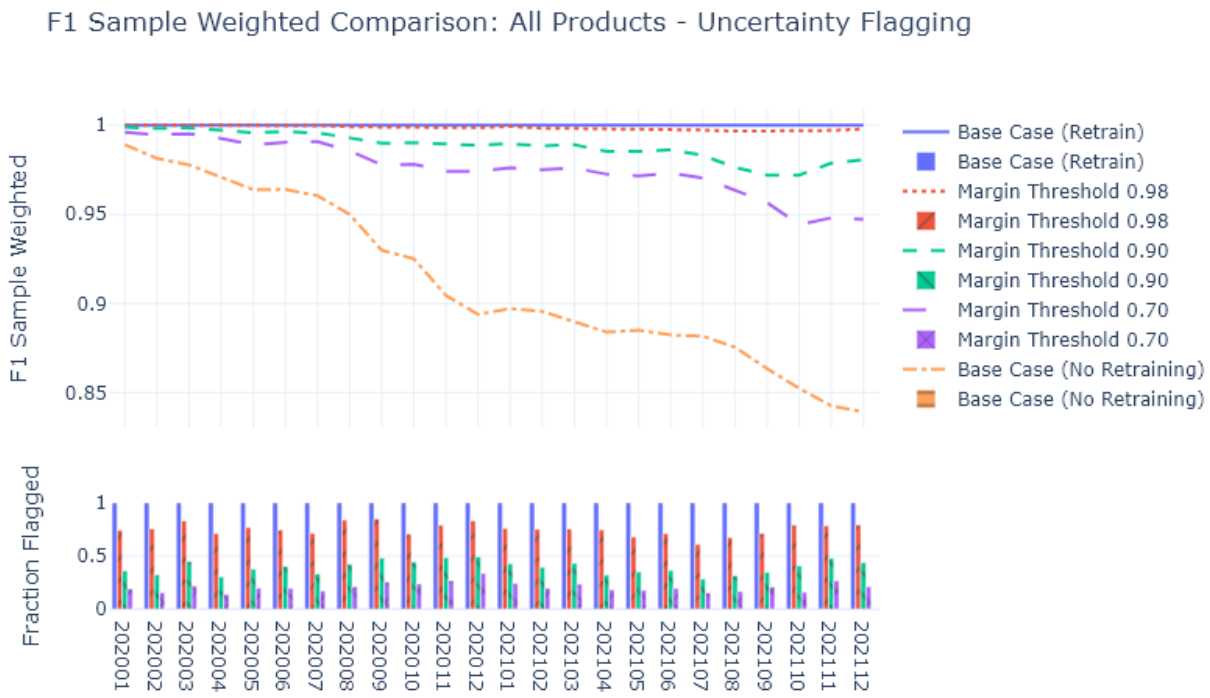
To analyze the effect of uncertainty flagging we perform the same experiments as before, however with flagging products based on different thresholds of uncertainty. Products are flagged if their classifier uncertainty is below a given threshold. Margin threshold utilized in this study operates opposed to likelihood of the classifier being correct – in other words increasing the threshold will lead to more products being flagged as products the classifier is more "confident" in will be flagged. Similar to random flagging, we again compare the performance of the classifier on new products for different thresholds (Figure 16). We find that using products flagged using classifier margin is quite efficient as refitting the model every 3 months on margin flagged data helps to prevent degradation in the classifier due to drift. We observe that model performance did drop in the final three-month period, which could be an indication that margin flagging is introducing bias into the training data and the model, a topic that needs more study to fully validate. We also find that increasing the margin threshold flags more products but does not improve the classifier performance as dramatically as the changes observed with the random flagging technique.

Figure 16: Comparison of Classifier Performance (with re-fitting) for Different Margin Thresholds



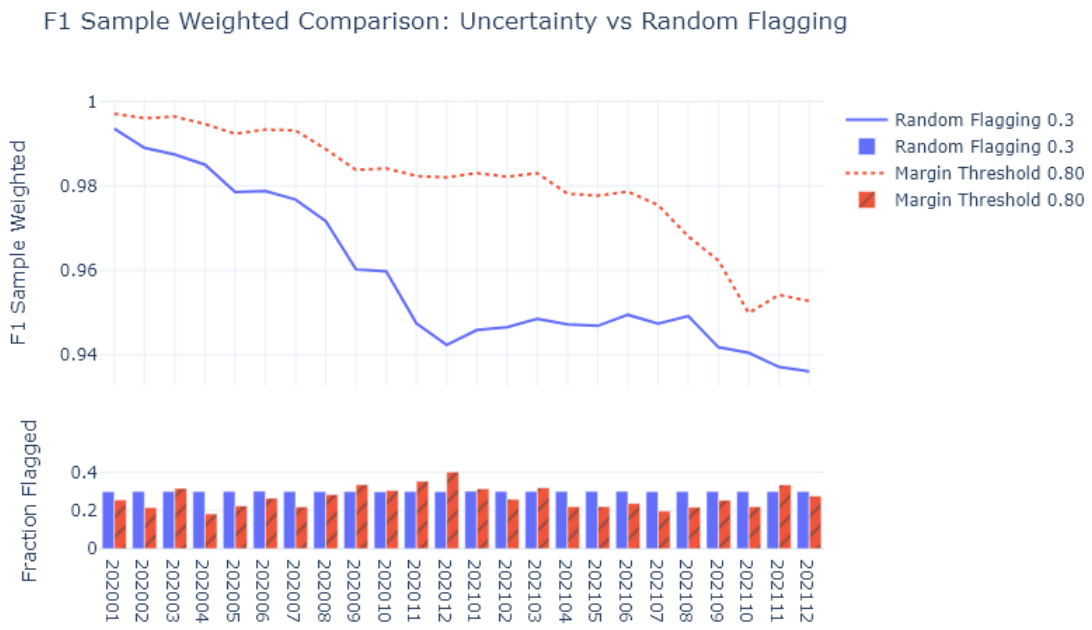
As for the random flagging outlier method, we compare the impact of the uncertainty outlier method on all products in the sample (Figure 17). We show the F1 score degradation over time for all products observed in each reference period post QA, with a sub-plot showing the fraction of products flagged in each month in order to show how much effort is needed by NSOs if they choose this promising method. Though not constant, each threshold tended to flag a similar proportion of products each month, with higher thresholds flagging more products (Figure 17).

Figure 17: Comparison of Post QA F1 Score for Different Margin Thresholds



Contrasting uncertainty flagging with random flagging, uncertainty flagging was better at identifying misclassified products compared to randomly flagging new products. With uncertainty flagging, a higher fraction of errors was flagged, for the same amount of review effort. To reach this result, we flag approximately 30% of new products for review each period using a margin threshold of 0.80; this is compared to randomly flagging 30 percent of new products (Figure 18).

Figure 18: Comparison of Post QA F1 for Uncertainty vs Random Flagging

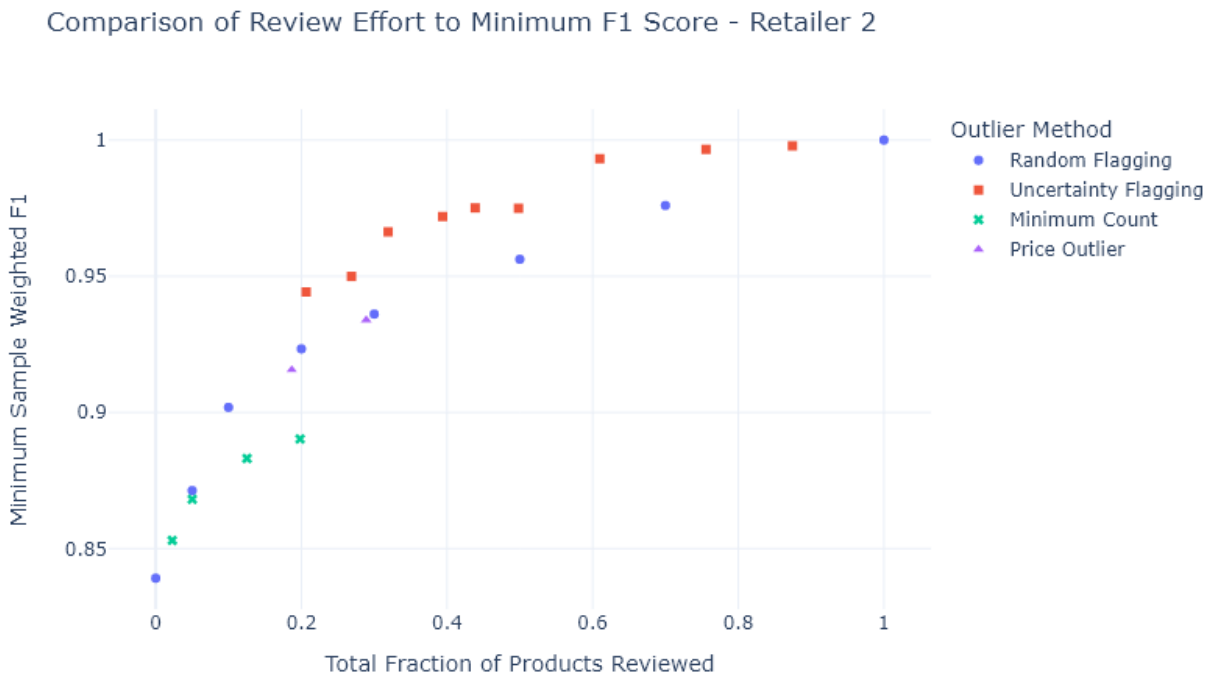


3.4.3. Comparison of Flagging Methods

Compared to confidence or random flagging methods, conceptually, flagging based on price outliers and category counts are more likely to be designed to support mitigation of misclassification on the final index calculation, than to utilize for retraining models. However, we compared these methods to evaluate how well classifier performance improved if these methods were also trialed individually for ML model retraining. We observed a similar impact on overall accuracy compared to random flagging, for a comparable proportion of new products flagged each reference period.

We evaluate all four different flagging methods by comparing the F1 score to the total fraction of new products flagged for review. The minimum sample weighted F1 score, for all reference periods in the 24-month production window is plotted on the y axis, compared to the fraction of new products that are flagged on the x axis (Figure 19). In general, flagging a higher percentage of new products improves the overall class F1 score, for the same method. Methods with higher F1 score at lower fraction of products flagged, are comparatively more efficient at finding misclassifications. Expansion of findings such as these, NSOs can estimate either the labelling effort required to achieve a minimum F1 score, or the F1 score that can be expected given a fraction of products flagged, for each method. Ultimately it will be up to the NSOs to balance the trade-off between classification performance and labelling budget when implementing a flagging and review protocol in production.

Figure 19: Empirical Performance of Different Flagging Methods



4. Discussion

This research has shown through an empirical case study that misclassification errors are important to consider by focusing on the key aspects of applying ML into production. Firstly, as labelling creates datasets that are used for ML model training or re-training, designing a robust process is key, as annotation processes can be expensive from a resource point of view. Our findings validate and build on previous studies (Greenhough, Martindale and Sands 2022), showing that while annotators are overall quite consistent, additional resource allocation could be directed towards categories known to be more heterogeneous and subjective compared to other categories that need less investment. For instance, classes where human annotators traditionally perform well can potentially be annotated or reviewed by a single individual, whereas more difficult classes should be reviewed by multiple annotators. Furthermore, understanding the categories for which disagreement exists between human annotators could be utilized to improve the class hierarchy over time, such as by modifying class definitions to remove ambiguity.

Even once high performing models are trained on high quality data, misclassification will still occur. Hence our experiments on the impact of misclassification on a representative elementary aggregate in one reference period, or over time on one elementary index, provide justification for utilizing high performance classifiers and validating predictions in production. Specifically, results from the misclassification experiments show that the introduction of misclassifications can introduce both bias and variance into the underlying index calculation. Furthermore, simulated misclassifications demonstrated that though both precision and recall are important metrics; maximizing class precision, is most important for minimizing bias in the index for a specific class. These findings could be useful in designing a review process for production, allocating review efforts to ensuring high class precision for the most important, in scope, classes. Additionally, results from our representative use of a price index showed that while the index was tolerant to moderate misclassification, it still deviated from the true index.

Further research is needed however to evaluate sensitivity of multiple price index methods to varying levels of misclassification and provide a comprehensive picture.

Given the impact of higher levels of misclassification on a price index, minimizing model degradation over time is key. Our results demonstrate that model degradation over time occurs, however it can be addressed through periodic refitting, using new products that have been reviewed. In our experiments, even refitting every three months, adding at least 10 percent of new products to the training dataset, was sufficient to obtain classifier performance similar to the base case of refitting with 100 percent of new products. While these findings are specific to our dataset, and may not generalize perfectly, they should serve as a useful guide for other similar datasets. At the same time, in the absence of quality control, the quantity of misclassified products will gradually increase over time as new products enter the sample and old products exit. Flagging and reviewing products are required to correct misclassifications, evaluate model performance, and refit the classifier model. In our experiments we have the benefit of knowing the true class of all products; in a production setting this is not the case; the true class is only ever known for those products that are flagged and reviewed by a human annotator.

Once a validation process is utilized to quality control the prices index and utilize this dataset as feedback to retrain the model, NSOs face a question of which outlier method is most applicable to utilize in a retraining dataset. Our experiments showed that model uncertainty was effective at catching many misclassified products and outperformed random flagging for use in retraining models. At the same time, uncertainty flagging may introduce bias into the classification model, if exclusively used for model refitting, a topic that is outside the scope of this research and needs to be evaluated in more detail. Furthermore, products flagged using model uncertainty are not appropriate for an unbiased estimate of model performance. As such, we recommend to additionally flag a portion of the dataset randomly to use for unbiased model evaluation.

Additional flagging methods may be appropriate, depending on operational requirements. Our experiments tested flagging based on product count and price outliers. Though experiment results suggested that these methods were no better at detecting misclassified products for model retraining purposes, these methods should still be part of a robust flagging strategy in a production setting, flagging products that are most likely to be impactful on the final calculated index.

While the specific flagging methods tested in this work may not provide complete evaluation of flagging methods, we hope that the approach and framework developed will prove a useful starting point for future work. Additional development of price flagging, as well as consideration for weights, in scanner retailers are some specific areas for future work. In the future, a wide array of different flagging approaches can be considered; it is important to keep in mind that no single flagging method will be capable of meeting all production needs and a combination of different methods will likely be required.

5. Conclusion

While the research contributes to the literature on misclassification within consumer price indices, there are several limitations worth noting that could guide further research on the topic. Firstly, the authors are not aware of a theoretical framework for the analysis of the impact of misclassification on price indices calculated using alternative data sources. Furthermore, a comprehensive study showing how each price index or extension method is sensitive to misclassification could support the conversation on index choice. Secondly, an expansion of the research to scanner data and longer time horizons would benefit the understanding of how misclassification could build over time and how other variables, such as product weight, could be utilized as part of outlier detection when attempting to mitigate misclassification. Thirdly, as NSO adoption of alternative data means that the volume

of new products would be quite high and require considerable resources to maintain manually, a targeted study on how multiple outlier flags could be combined in an optimal way to balance investment into validation and index accuracy would be beneficial to demonstrate how production processes could be designed. Finally, we note that the retailers studied in this work had moderately homogeneous product lines. We expect that the impact of product misclassification will be even more important for retailers with vast product lines, for example department store or big box retailers. In addition, there exists the possibility of other unpredictable shocks and disruptive events, such as website re-design, which should be considered in the design of a resilient production system.

Bibliography

Advisory Panel on Consumer Prices – Technical. 2019. *Guidelines for selecting metrics to evaluate classification in price statistics production.* Technical report, UK Statistical Authority. https://uksa.statisticsauthority.gov.uk/wp-content/uploads/2019/08/APCP-T1910-Classification-metrics-guidelines.pdf.

Artstein, Ron. 2017. "Inter-annotator agreement." In *Handbook of linguistic annotation*, edited by Nancy Ide and James Pustejovsky, 297-313. Dordrecht: Springer Netherlands. doi:10.1007/978-94-024-0881-2_11.

Bayram, Firas, Bestoun S. Ahmed, and Andreas Kassler. 2022. "From concept drift to model degradation: An overview on performance-aware drift detectors." *Knowledge-Based Systems.*

Chessa, Antonio G. 2021. *Extension of multilateral index series over time: Analysis and comparison of methods.* Technical report, Department of Consumer Prices, Statistics Netherlands.

Choi, InKyung, Andrea del Monaco, Eleanor Law, Shaun Davies, Joni Karanka, Alison Baily, Riitta Piela, et al. 2022. "ML Model Monitoring and Re-training." ML 2022 Model Re-training Theme Group, UNECE. https://statswiki.unece.org/download/attachments/338329823/ML2022%20Model%20Retraining%20Report.pdf?version=2&modificationDate=1673345538557&api=v2.

De Waal, Ton. 2013. "Selective Editing: A Quest for Efficiency and Data Quality." *Journal for Official Statistics* 473-488.

De Waal, Ton, Pannekoek, Jeroen, and Sander Scholtus. 2011. *Handbook of statistical data editing and imputation.* Vol. 563. John Wiley & Sons.

Dongmo-Jiongo, Valéry. 2021. "Innovative uses of web scraped data in the Canadian Clothing and Footwear Consumer Price Index." *High Level Group on the Modernization of Official Statistics: Machine Larning 2021 Monthly Meeting Grou.*

Eurostat. 2022. *Guide on multilateral methods in the Harmonised Index on Consumer Prices (HICP) — 2022 edition.* Manual, Luxembourg: Publications Office of the European Union.

Eurostat. 2017. *Practical Guide for Processing Supermarket Scanner Data.* European Commission.

Fox, Kevin J., Peter Levell, and Martin O'Connell. 2022. *Multilateral index number methods for Consumer Price Statistics.* ESCoE Discussion Paper 2022-08, Economic Statistics Centre of Excellence.

Gama, Joao. 2013. "A Survey on Concept Drift Adaption." *ACM Computing Surveys* 44.

Greenhough, Liam, and Mario Spina. 2022. *Outlier detection for rail fares and second-hand cars dynamic price data.* Office for National Statistics. https://www.ons.gov.uk/economy/inflationandpriceindices/methodologies/outlierdetectionforrailfaresandsecondhandcarsdynamicpricedata.

Greenhough, Liam, Hazel Martindale, and Helen Sands. 2022. "Modernising the measurement of clothing price indices using web-scraped data: classification and product grouping." *17th Meeting of the Ottawa Group.* Rome, Italy.

Harms, Alexander, and Siemen Spinder. 2019. *A comprehensive view of machine learning techniques for CPI production.* Statistics Netherlands.

HLG MOS. 2019. "Generic Statistical Business Process Model (version 5.1)." https://statswiki.unece.org/display/GSBPM/GSBPM+v5.1.

Hov, Kjersti Nyborg. 2021. "Machine learning in the Norwegian CPI: A classification tool." *Group of Experts on Consumer Price Indices.* online. https://unece.org/sites/default/files/2021-05/Session_1_Norway.pptx.

Huyen, Chip. 2022. *Designing Machine Learning Systems.* O'Reilly Media, Inc.

Manual, Consumer Price Index. 2020. *Concepts and Methods.* Geneva: ILO/IMF/OECD/Eurostat/UNECE/The World Bank, International Labour Office (ILO).

Martindale, Hazel, Edward Rowland, Tanya Flower, and Gareth Clews. 2020. "Semi-supervised machine learning with word embedding for classification in price statistics." *Data & Policy* 2 (e12).

Meertens, Q. A., Diks, C. G. H., H. J. Van den Herik, and F. W. Takes. 2020. "A data-driven supply-side approach for estimating cross-border Internet purchases within the European Union." *Journal of the Royal Statistical Society Series A: Statistics in Society* 183 (1): 61-90.

Moreno-Torres, Jose G., Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. 2012. "A unifying view on dataset shift in classification." *Pattern recognition* 45 (1): 521-530.

Myklatun, Kristian Harald. 2019. "Utilizing Machine Learning in the Consumer Price Index." *28th Nordic Statistical Meeting, Helsinki.*

Office for National Statistics. 2020. "Automated classification of web-scraped clothing data in consumer price statistics." https://www.ons.gov.uk/economy/inflationandpriceindices/articles/automatedclassificationofwebscrapedclothingdatainconsumerpricestatistics/2020-09-01.

Oyarzun, Javier, and Laura Wile. 2022. "Quality Control of Machine Learning Coding: A Statistics Canada Experience." *UNECE.*

Piela, Riitta. 2021. "From Theory to Practice: Detecting Model Decay (or a journey to better understanding of MLOps)." *ONS-UNECE Machine Learning Group 2021 webinar.* https://statswiki.unece.org/download/attachments/330367795/Finland_From%20Theory%20to%20Practice.pdf?version=1&modificationDate=1637319706255&api=v2.

—. 2022. *Work Stream 4 - Model Retraining.* HLG MOS, Machine Learning Group 2021. https://statswiki.unece.org/download/attachments/293535864/ML2021_WS4_Finland.pdf?version=1&modificationDate=1643981040799&api=v2.

Platt, John C. 2000. "Probabilistic outputs for SVMs and comparisons to regularized likelihood methods." In *Advances in Large Margin Classifiers*, by Alexander J. Smola, Peter J. Bartlett, Dale Schuurmans and Bernhard Schölkopf, 61-74. Cambridge, Massachusetts: MIT Press.

Scholtus, Sander, and Arnout van Delden. 2020. *On the accuracy of estimators based on a binary classifier.* Discussion Paper, CBS.

Sculley, David, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. 2015. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems* (28).

Settles, Burr. 2009. *Active learning literature survey.* Madison: University of Wisconsin-Madison, Department of Computer Sciences. http://digital.library.wisc.edu/1793/60660.

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

Valliappa Lakshmanan, Sara Robinson, Michael Munn. 2020. *Machine Learning Design Patterns.* O'Reilly Media.

van Delden, Arnout, Sander Scholtus, and Joep Burger. 2016. "Accuracy of mixed-source statistics as affected by classification errors." *Journal of official statistics* 32 (3): 619-642.

Van Loon, Ken. 2020. *Scanner data and web scraping in the Belgian CPI.* National Academies. https://www.nationalacademies.org/documents/embed/link/LF2255DA3DD1C41C0A42D3BEF0989ACAECE3053A6A9B/file/D124958ED038610E68986C71BEC8EA6D97CBF5F39C35?noSaveAs=1.

Widmer, Gerhard, and Miroslav Kubat. 1996. "Learning in the Presence of Concept Drift and HIdden Contexts." *Machine Learning.*

Yung, Wesley, Siu-Ming Tam, Bart Buelens, Hugh Chipman, Florian Dumpert, Gabriele Ascari, Fabiana Rocci, Joep Burger, and InKyung Choi. 2020. "A quality framework for statistical algorithms." *Statistical Journal of the IAOS* 38 (1): 291-308.