Clothing Price Index using Web-Scraped Data

Ahmet Yusuf Aydin, Steven Jones, Laura Christen (Office for National Statistics)
ahmet.aydin@ons.gov.uk

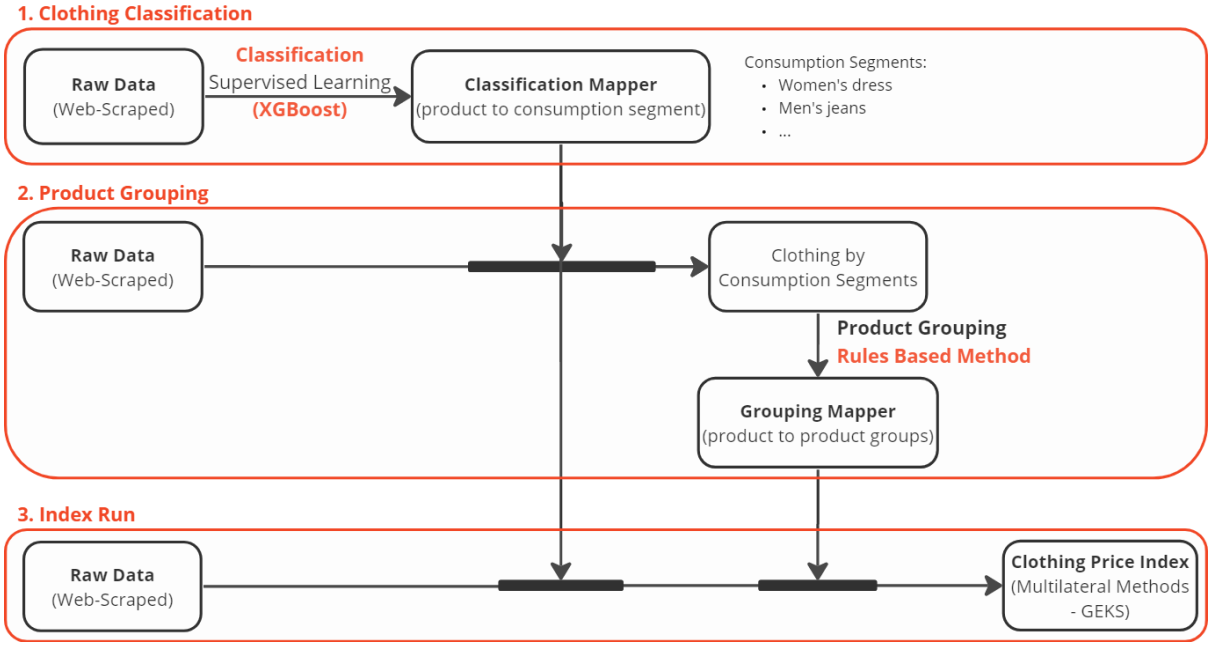
1. Introduction

The transformation of UK Consumer Price Statistics programme (ONS, 2022) aims to incorporate alternative data sources including scanner and web-scraped data into the production of major consumer price statistics. We have developed new methods to process the data and integrate into the production of Consumer Price Index (CPI).

Clothing contributes approximately 5% of the CPI basket in the UK and currently is covered with manually collected data. We obtain web-scraped data from the online shopping websites of main retailers in the clothing sector. We aim to increase product coverage with the high numbers of clothing items collected via web-scraping compared to manual price collection. This helps us to have more representative price data as we collect daily prices and improves granularity of the index since we can cover more various types of clothing.

We process web-scraped textual data using Natural Language Processing (NLP) and machine learning techniques to build a clothing price index. This paper outlines three of the key pipelines[1] we use to build the index as shown in Figure 1.

Figure 1. The structure of the project pipelines



First, the classification pipeline produces a classification mapper which maps individual clothing products to narrowly defined clothing consumption segments such as “women’s dresses” or “men’s jeans”. We build a supervised machine learning model using a gradient-boosted tree algorithm (XGBoost) to classify web-scraped clothing data. We train the model to learn classification rules with a human labelled dataset created by price experts within ONS. Our classifier performs well with an F1 score of 85% on average, while precision is over 90% on some consumption segments. Then, we apply the model to the data we receive to predict classes for new data. We use the classification mapper in

[1] In addition to these three pipelines, there are additional pipelines such as outlier detection and aggregation that are out-of-scope for this paper.

both the product grouping and index run pipelines to read the raw data by consumption segments and produce a price index at the granular level of consumption segments.

Secondly, we create a product grouping mapper which maps each product to a product group using a rules-based method. The pace that products enter and exit clothing markets makes it challenging to appropriately measure their changing prices. The aim of product grouping is to group similar products and track average prices of each (homogeneous) group instead of individual products to increase product match over time. This is crucial for the clothing price index due to the high churn with high product turnover rates and seasonality in the market.

Thirdly, we create a clothing price index using multilateral index methods as they allow better use of the dynamic structure of web-scraped data with entering and leaving products. We utilise the classification mapper and product grouping mapper created in the previous stages while building the clothing price index. The classification mapper helps us to build price indices at a granular level of consumption segments and aggregate them to obtain a single index for clothing. The product grouping mapper enables us to overcome the product churn problem and run the index with continuous price data for a high proportion of products.

In conclusion, this project will allow us to modernise UK consumer price statistics by making better use of new data sources and innovative methods.

2. Data

We obtain web-scraped clothing data since June 2020 from 17 online retailers in the UK which covers around 1000 brands. This makes more than 900,000 unique clothing products in each month, extending our coverage significantly compared to the traditional manual data collection which covers approximately 20,000 products. Working with such a different nature of data at this scale requires extensive data processing and implementing new innovative methods for index calculation. The next section outlines the research and implementation of these methods.

3. Flow of Pipelines

3.1. Classification

The introduction of alternative data sources means that, each month, we are web scraping prices of more than 900,000 unique clothing products from some of the UK's largest clothing retailers. These unique products need to be classified into relatively homogeneous groups, which we describe as consumption segments. Given the size of this dataset, manually classifying these products would not be feasible, so we are investigating automating the process using supervised machine learning.

In this section we will discuss:

- How we define consumption segments
- Creating a labelled dataset
- Choosing a classification model
- Methods of model improvement

3.1.1 Introducing the consumption segments

The Office for National Statistics (ONS) currently use the international classification structure, the Classification of Individual Consumption according to Purpose (COICOP), to produce internationally comparable inflation aggregates. However, to enable these comparisons, the COICOP structure is relatively broad. For example, the lowest level of COICOP might be "women's garments", a large and heterogeneous grouping of all clothing garments worn by women.

To calculate indices at a more granular level, we will use lower-level "consumption segment" aggregates (ONS, 2021), which are unique to the UK, along with some category-specific extra strata levels. For clothing, we use a combination of age, gender, and clothing type to define each

consumption segment. For example, the "women's garments" COICOP category may contain consumption segments such as "women's dresses", "women's leggings", "women's t-shirt/crop top" and a range of other women's garments deemed representative of consumer spending. The consumption segments enable us to produce indices for groups of similar products, which can then be aggregated into a higher-level index.

Defining consumption segments requires finding a balance between being relatively homogeneous in composition, simple to classify, and of a large enough size to produce reliable statistics. For example, a consumption segment of "women's tops" would be easy to classify and large enough to compute reliable indices, but the products allocated to this segment could be varied, including tops of all styles and for all occasions. We could define a narrower segment of "women's formal blouses", but this could result in a small and relatively ambiguous segment, as what defines a "formal blouse" is not always clear cut. The classifier may not be able to confidently predict membership of this segment, leading to low performance and inaccurate indices. We need consumption segments with high enough classification performance to produce reliable and unbiased indices, whilst also maximising homogeneity.

3.1.2 Creating a labelled dataset

Supervised machine learning uses a human-labelled dataset, based on a sample of the data, to learn rules which are then applied to further data. Machine learning algorithms often perform better with more data. Therefore, we needed a process to obtain a large, high-quality, labelled dataset. To produce a sample of products to be labelled, we used stratified sampling on the retailer hierarchy with weights proportionate to the number of consumption segments in each age and gender group. For example, if 26% of our segments are men's clothes, we require 26% of the sample data to come from hierarchies related to men's clothing. We labelled this sample using a bespoke labelling application developed in-house. We have currently labelled 162,700 products.

However, there can often be an element of subjectivity when manually classifying a dataset. For example, the term "sweater" can often be used by retailers interchangeably for both sweatshirts and jumpers, leading to labellers sometimes being inconsistent in how they place this product. Because of this we took steps to understand how subjectivity affected our labelling results by re-labelling a proportion of our data. We found that there was a broad consensus on how to label most products, with labellers being consistent in 88.8% of cases. This consistency level varied from class to class, with some classes being highly consistent and others being more problematic to classify. This information is useful because it will give us a benchmark for classifier performance, which we can use to set an expected performance limit for the machine learning algorithm.

3.1.3 Classification model

For our classification model, we explored several machine learning algorithms including decision trees, logistic regression, random forests, support vector machines, and XGBoost. See ONS (2020a) and Martindale et al (2019) for a more detailed comparison of model results. Our preferred algorithm is XGBoost, an ensemble model which sequentially trains numerous gradient-boosted decision trees. Each tree is trained to improve on errors made by previous trees, and the final prediction of the model is the weighted sum of all predictions made by previous tree models.

XGBoost is our preferred algorithm for several reasons. In our research, we obtained the highest performance metrics (in macro-averaged precision, recall and F1 scores) with XGBoost and Support Vector Machines. However, XGBoost had practical advantages, having faster training times (with GPU support) and the ability to provide confidence scores which enabled us to evaluate how confident the algorithm was in predicting each class. We may use these scores for confidence thresholding, which we will discuss in the next section.

3.1.4 Model improvement

3.1.4.1 Confidence threshold

For this section, we consider a few metrics. Precision measures the proportion of positive predictions that are correct, and recall measures the proportion of actual positives correctly identified. A third metric, the F1-score, measures a harmonic average of precision and recall, whereas the F0.33-score measures a harmonic average where precision is considered three times more important than recall. When measuring inflation, we may prefer precision over recall since, for example, capturing every single "women's dress" is of a lesser priority than being certain that the "women's dresses" chosen to represent inflation are in fact women's dresses.

When predicting the membership of a product to a consumption segment, the XGBoost algorithm provides a confidence score that the product belongs to each class. For example, it might classify a product as "women's sports top" and give a 60% confidence that this is its true class. Allowing the algorithm to make predictions with low confidence scores may cause the algorithm to make more erroneous predictions (reducing precision) whilst capturing more cases overall (increasing recall).

Therefore, to increase the precision of the model, we are exploring the implementation of a "confidence threshold" which requires that the algorithm give a probability above a certain threshold before allocating a product to a class. For example, setting this threshold to 0.8 would require that the model be 80% confident that a product belongs to the chosen class before classification. If the probability is below this threshold, it would give "no prediction". The macro precision, recall, F1 and F0.33 scores for different confidence thresholds are shown in Table 1, below.

Table 1. Macro Precision, Recall, F1 and F0.33 Scores of XGBoost algorithm on test data, using different confidence thresholds.

Threshold	Precision	Recall	F1 Score	F0.33 Score
None	0.86	0.84	0.85	0.86
0.70	0.91	0.69	0.77	0.88
0.75	0.92	0.66	0.75	0.89
0.80	0.92	0.61	0.72	0.88

These results demonstrate that increasing the confidence threshold improves the precision of the model at the expense of recall. We may implement this if we decide that high precision is more important than high recall. For example, if precision is considered three times more important than recall, optimal results for the F0.33 are found with a threshold of 0.75.

3.1.4.2 Confusion matrix

We find that the consumption segments which are easiest to classify often contain products with an indicative word in their product name. For example, most "jeans" products contain the word "jeans" in their descriptor. This means these segments generally have high precision and recall scores. However, there are classes that the algorithm struggles to classify because they have words which overlap with other segments. For example, sports clothes such as "sports shorts" often have similar descriptors to non-sports clothing like "men's shorts". This generally leads to labeller inconsistency, misclassification and lower performance scores for these classes. Determining which classes the model is struggling to distinguish between would help us to pinpoint improvement areas for the model.

To this end, we have produced a confusion matrix which establishes which consumption segments the classifier is struggling to predict accurately. It does this by comparing the predicted value to the actual value for each class, therefore showing us points of contention. Table 2 presents some classes which the algorithm struggles to accurately classify, and the class that they are most confused with.

Table 2. Example of classes which are inaccurately classified, and their points of contention

Class	Point of Contention
Girls' sports top	Girls' top/t-shirt/crop-top
Boys' outfit set	Boys' full tracksuit
Men's sports top	Men's t-shirt
Women's sports top	Women's top/t-shirt/crop-top
Boys' vest	Boys' t-shirt

As discussed previously in Section 3.1.2, we need to define consumption segments which ensure a satisfactory trade-off between homogeneity and class performance. We could, for example, combine "men's sports tops" with "men's t-shirt", raising the performance of the consumption segments but resulting in a less homogeneous grouping of products. To help us make this decision, we calculated three metrics to aid decision-making: the weight of the class, the F1 score, and the change in F1 score effected by combining the class with its contending class. From this we have identified around 20 classes which either have very small weights, low F1 scores, or both. We may decide to combine some of these classes if we find that the F1 score improves and the segment remains sufficiently homogeneous.

3.1.5 Classification Mapper

Once we create a high-performing classification model, we can apply the same model to new data to get a classification mapper which maps each product (along with all their associated web scraped prices) to a consumption segment. We could potentially create separate price indices for each consumption segment and aggregate them to obtain a single clothing price index. However, the extremely dynamic nature of the clothing market, with fast product entry and exit, causes problems for the way our indices are calculated. Prior to calculating indices, we are therefore looking to use product grouping.

3.2. Product Grouping

Typically, National Statistical Offices use matched-model price index methods to measure inflation. Matched-models compare the prices of the same set of products in two different months. Products that enter or leave the market between the two months being compared will lack a price in one of the two months and therefore cannot be used. In the case of clothing, this can cause the index to become unrepresentative due to the rapid product entry and exit. Clothing products rarely stay in the market for a full year as seasons and fashion trends change several times a year. Therefore, "product churn" is a fundamental problem for the clothing index.

Traditional methods of calculating our Consumer Price Index (CPI) requires finding a substitute product when a product leaves the market. This replacement is a manual step in traditional methods; however, introducing scanner or web-scraped data to the CPI calculation complicates this process as these manual processes are not viable with the huge volume of products and prices data.

The product grouping pipeline aims to group products which are similar or substitutable from the customer perspective. We reduce the effect of the product churn problem by tracking average prices of groups instead of individual products, since product groups are more likely to survive even though some products within group are leaving the market.

Product groups should be large enough to control for product churn. In other words, we try to capture a high proportion of products in the index calculation by making broad product groups to survive through a year despite losing some of its constituents. We measure this by match rate which is the proportion of matching groups from the base period to current month. On the other hand, those groups should also be homogeneous in terms of quality and from a customer perspective so that compositional effects do not bias inflation. These two criteria compete as we need to have finer

groups for the sake of homogeneity, but match rate decreases with finer groups. Therefore, product grouping should have a balance between match rate and homogeneity.

3.2.1 Assessment Measure: MARS Score

Chessa (2019) introduced the "Match Adjusted R Squared" (MARS) score to assess the success of groups as a product of match rate and homogeneity:

$$\mathrm{MARS}_t = \mathrm{R}_t \, \mu_t$$

where μ_t is the match rate and R_t is the R-squared measuring in-group price similarity within the current month.

Match Rate: Product match in the absence of product grouping is the share of products in the current month matching with the base period. Once we form the groups, match rate is the share of matching groups from the base period to current month within all product groups in the current month. We can cover a wider range of products in index calculation as more products included in the groups which match the groups in the base period.

Homogeneity: R-squared is a measure of in-group price similarity within the current month. It measures the proportion of explained variance in prices by grouping relative to the total variance in prices without grouping.

There are two caveats of this measure. First, the original R-squared within MARS weights these variations with quantities. However, we use an unweighted version as web-scraped data do not include quantity or expenditure information.[2] Secondly, MARS measures homogeneity only with price similarity. We cannot measure homogeneity in terms of purpose or quality as we cannot quantify them with web-scraped data.[3]

3.2.2 Rules Based Method

We need a set of rules for each consumption segment to assign individual items to product groups. We search through the attribute columns in the data to find these rules. As the data scraped from the retailer websites are textual and unstructured, we process attribute columns to find key words to use as rules forming product groups.

There are alternative ways to determine those key words. They should reflect the characteristics and quality of the products to form homogeneous groups. Ideally, we can select such key words with a visual inspection of the data and using domain knowledge on the market. However, this is not practical for each consumption segment due to the scale of the data and lack of resources and expertise. Therefore, we propose an automated approach.

The first approach we consider is to derive the most commonly occurring words in each attribute column. We use NLP techniques to clean and process textual data. We remove punctuation and numbers, remove stop words, and standardise retailer and brand columns to account for different versions of same retailer/brand. We also remove some common words which do not differentiate products within consumption segment and do not provide any quality information, such as *"dress"* for *"women's dresses"* consumption segment.

Once we clean and process the data, searching for a fixed number of most frequent words in each attribute column gives us a rule dictionary. For each product, we flag the words in the rule dictionary if they appear in the attribute columns. The combination of those flag words creates a group identifier

[2] There is separate research within ONS on expenditure proxies to estimate expenditures with product characteristics using web-scraped data. (ONS, 2020c)

[3] We consider using human evaluation of product similarity within groups.

for each product and assigns them to a particular group. Table 3 provides a simplified example of rules-based product grouping with two attributes and two rules for each attribute.

Table 3a. Rules dictionary with two attributes and two rules for each attribute

	Rules Dictionary	
Attributes:	**Product Name**	**Material**
	v-neck	polyester
	maxi	cotton

Table 3b. Rules-based product grouping with rules dictionary provided in Table 3a

	Product Name	Material	Group identifier
Product 1	v-neck dress	polyester	v-neck_polyester
Product 2	floral maxi dress	100% cotton	maxi_cotton
Product 3	white maxi dress	cotton elastic	maxi_cotton

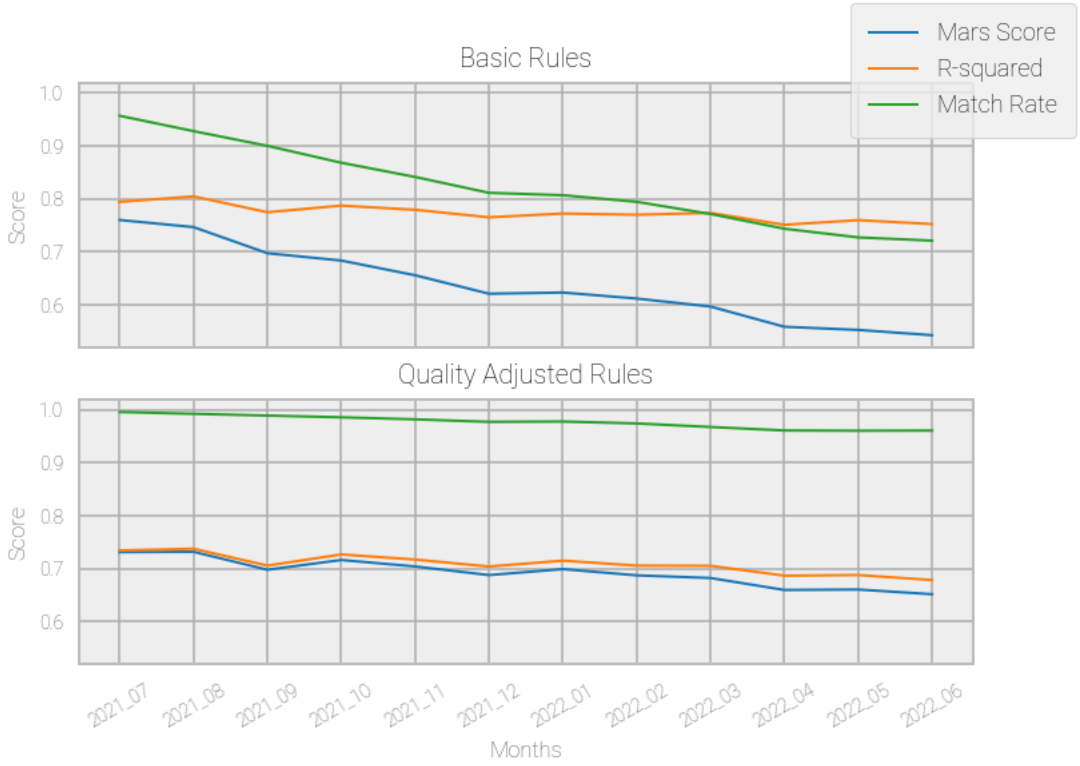
3.2.2.1 Quality Adjustment

The basic approach to form the rules dictionary naively ranks the words according to their frequency in each attribute column. However, some key words with less frequency might have greater importance to distinguish products. Therefore, we implement a quality adjustment to the rules dictionary to improve grouping performance.

First, we collect the hundred most frequent words in each attribute column. Then, we run a hedonic regression for each column with dummies for containing those words to quantify the impact of each key word on the price of a product. Re-ranking the words with statistically significant impact on price according to their contribution to price gives us a quality adjusted rules dictionary.

Figure 2 shows the impact of quality adjustment on MARS scores for product grouping with twenty rules from each attribute over a year. MARS scores increase significantly, especially towards the end of the period. We get significantly higher match rates, although R-squared decreases slightly.

Figure 2. MARS Scores with 20 Rules



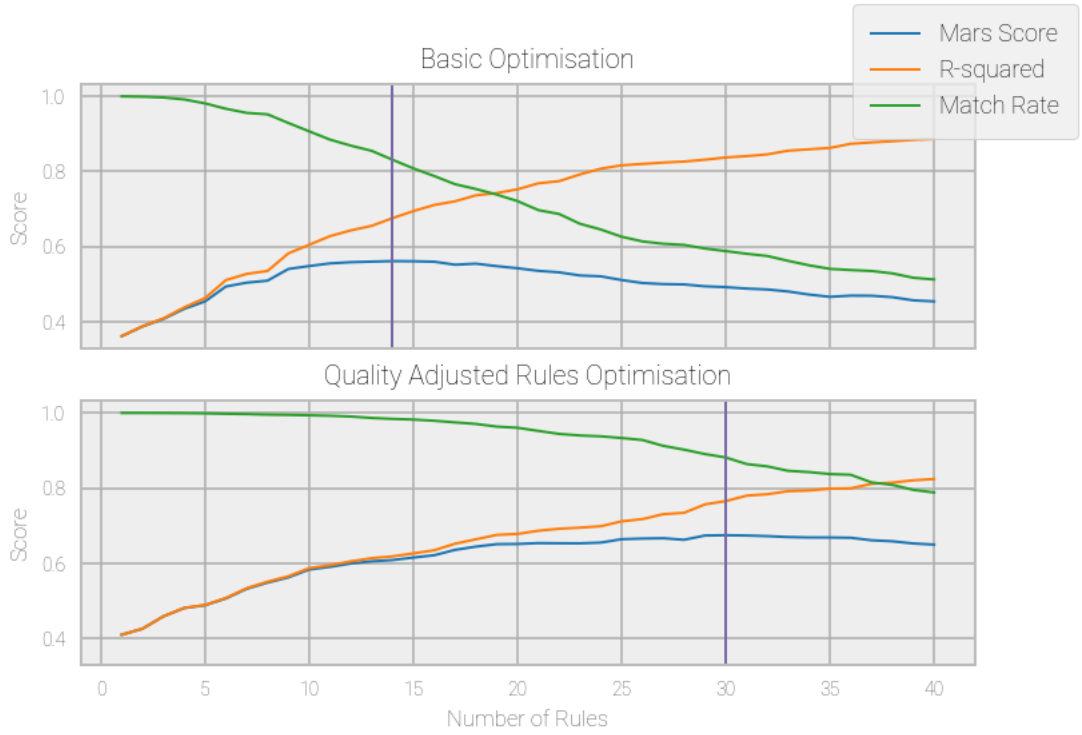
3.2.2.2 Rules Optimisation

The basic approach to rules-based product grouping takes a pre-determined fixed number of rules for each attribute column. However, we can find the optimal number of rules with an optimisation algorithm. The objective of optimisation is to reach a maximum MARS score for a better grouping outcome. We can maximise either the average MARS score over the assessment period or the MARS score in the latest month of the period. The second option is preferable as the MARS score tends to decrease over time with lower match rate towards the end of the period.

The basic method for finding the optimal set of rules for product grouping is to set a minimum and maximum number of rules, and to run grouping for each number of rules in between. The grouping with maximum MARS score will give the optimum number of rules. We can also use quality adjusted rules dictionary for this optimisation search.

Figure 3 shows the results of the basic optimisation with and without quality adjustment where we maximise the MARS score in the latest month of the assessment period. Match rates and MARS scores are significantly higher with quality adjustment as expected. The highest MARS score of 67.5% is achieved at thirty rules per column for quality adjusted rules.

Figure 3. MARS Scores in the latest month with number of rules from 1 to 40



We consider other alternative optimisation methods and research is ongoing for them.

A further improvement to the basic optimisation above is to treat the rules dictionary as a single set of words instead of taking each attribute separately and imposing the same number of rules for each column. We plan to implement a single hedonic regression on the set of most frequent words as a whole and have a single list of quality adjusted words. Then, we can run the optimisation by adding a single rule from quality adjusted list of words at each iteration.

The third option we are considering is a Hill-Climbing algorithm, which optimises over different combinations of words. This algorithm differs from the two methods above by partially not requiring any ranking of the candidate rules. After taking the 100 most frequent words from each attribute column as candidates, we can start with a minimum number of rules and find the next word with the highest contribution to MARS score by assessing each word in the candidate list. This algorithm is able

to consider cases where more impactful rules are ranked lower and have higher contribution to the MARS score in a combination of previously added words.

Our fourth option to optimise product grouping problem is using a Genetic Algorithm which takes inspiration from how the genomes of individuals in a species' population evolves over time by natural selection to produce fitter offspring.

In this instance the process begins by dividing the data by retailer and finding some number of most frequent words per attribute, and subsequently constructing a Boolean-valued attribute matrix for the retailer data. Then a population of specified size is initialised where each genome is a random sequence of 1s and 0s representing the inclusion or exclusion of a candidate rule in the ruleset.

The selection, crossover and mutation stages ensure respectively that genomes giving higher MARS scores are more likely to be selected for reproduction, that offspring can feature traits from two genomes and that gene values not present in the initial population can be expressed by future generations.

The method has been adapted to optimise the rulesets for each retailer separately before assigning products to product groups and calculating the MARS score and adapted to make use the of the quality-adjusted rules.

As the methods are still in development, their performances have not yet been assessed fully. In the case of optimising one set of rules for all retailers, using the most common words as candidate rules, using all the data collected a MARS score of 0.700 was achieved after allowing each optimisation to run for 50 generation. For optimising retailer-by-retailer, using the quality-adjusted rules with a 10% sample of the same data as above a MARS score of 0.782 was achieved after allowing each optimisation to run for up to 60 generations, but ending if no improvement was found in 12 consecutive generations.

In contrast with the methods above the Genetic Algorithm does not use a ranked list of words, it considers all candidate rules equally, which eliminates the risk of bias from the choice of ranking quantity. However, the method is much more computationally expensive.

3.2.3 Grouping Mapper

Once we run product grouping, each item will have a group identifier which indicates the group it belongs to as in the example in Table 3. We use this grouping mapper in the index run pipeline to calculate average prices within groups each month and track those average prices in time for our index calculation.

3.3. Index Run

Web-scraped data collected from retailers' websites are different in many aspects compared to traditional data sources. Apart from containing a wealth of information about the products and their attributes they are more frequent and cover a broader range of products. Therefore, they require more advanced index number and weighting methods.

Current index number methods used for the traditional data sources simply compare between two chosen time periods. It could be either fixed base by comparing prices of a product in each month of a year with a fix month like January, or chaining where monthly comparisons are chained to form an index series. These methods called bilateral methods as they compare prices of same products between two chosen time periods.

Multilateral methods, on the other hand, simultaneously make use of all data over a given time period. They allow better use of the dynamic structure of web-scraped data with entering and leaving products. Therefore, we plan to use multilateral methods in this project for calculating the clothing price index with web-scraped data.

There is a separate strand of research is going on within the Office for National Statistics (ONS) for multilateral index number methods (ONS, 2020b). The current advice is to use the GEKS-Jevons method in the absence of expenditure data. We can also use the GEKS-Törnqvist method using the product group sizes as quantities. Whilst group sizes do not truly reflect the real quantities sold for each product group, they may give a better approximation of the economic importance of each group, since it seems reasonable that a product group containing fifty products may be more likely to have higher sales than a product group containing five products.

We calculate price indices separately for each consumption segment and each retailer. We first aggregate retailers level indices to form an index at consumption segment level. Then, we aggregate consumption segment level indices to form a single clothing price index. We do not present the results here yet as they are very experimental, and we do not have full results yet.

4. Conclusion

Integrating web-scraped clothing data to index calculation is a part of the transformation of UK Consumer Price Statistics programme (ONS, 2022) with alternative data sources and innovative methods. The programme has achieved the first milestone in March 2023 by including rail fares transaction data in the production of official price statistics (ONS, 2023). We continue our research to improve methods to integrate clothing web-scraped data into the production of price statistics in a reliable and sustainable way.

References

Chessa, Antonio G. (2019) MARS: A method for defining products and linking barcodes of item relaunches - 16th Ottawa group Meeting

Eurostat Classification of individual consumption by purpose (COICOP)

Martindale, Hazel, Edward Rowland, Tanya Flower (2019) Semi-supervised machine learning with word embedding for classification - 16th Ottawa group Meeting

ONS (2020a) Automated classification of web-scraped clothing data in consumer price statistics - Office for National Statistics

ONS (2020b) New index number methods in consumer price statistics - Office for National Statistics

ONS (2020c) Using statistical distributions to estimate weights for web-scraped price quotes in consumer price statistics - Office for National Statistics

ONS (2021) Introducing alternative data into consumer price statistics: aggregation and weights - Office for National Statistics

ONS (2022) Transformation of consumer price statistics - Office for National Statistics

ONS (2023) Consumer price inflation, UK: February 2023 - Office for National Statistics