ADVISORY PANEL ON CONSUMER PRICES – TECHNICAL

Guidelines for selecting metrics to evaluate classification in price statistics production pipelines

Purpose

1. Even with well-trained machine learning (ML) classifiers on high quality data it is highly unlikely these classifiers are going to be able to get the classification of every price quote correct all the time. If we are to include these ML classifier in price statistics production pipelines, we must therefore understand how they might affect the output price index to minimise the impact of classifier errors on the final index. One aspect of this is to select appropriate metric(s) to measure and monitor classifier performance over time. There are numerous metrics to quantify classifier performance on a dataset, each with different properties for evaluating various aspects of classifier performance given certain conditions. This paper takes a theoretical approach to set out initial guidelines for how we should measure classifier performance in price statistics production pipelines to minimise error on the price index. This will help inform which classification methods are suitable for use, in the context of price index production pipelines using alternative data sources and provide direction for future development on this

Actions

2. Members of the Panel are invited to:
   a) comment on the suitability of these guidelines
   b) comment on direction of future work

Introduction

3. The Prices Alternative Data Sources project is developing a prototype pipeline to produce price indices from web-scraped and point-of-sale scanner price data. A crucial part of this pipeline is the development of methods to automatically classify individual price quotes according to the COICOP and ONS item level classification. There are numerous methods available to classify items to COICOP with each method capable of being optimised through hyperparameter tuning to a problem.
4. There are also numerous metrics that measure some aspect of how effective a classifier is working on a dataset. To assess different classification methods' performance in the pipeline and to monitor effectiveness over time, we need to decide which metrics to use for our particular data and purpose. There are numerous well-established metrics for evaluating classification performance. This paper evaluates a selection of the most commonly used metrics and summarises this in the guidance presented here for choosing an appropriate metric for building and evaluating classifiers in price index production pipelines, such that the classifier avoids biasing the final index.
5. This guidance provides an initial approach based upon theoretical considerations. As research and development of classification methods is working alongside development of these guidelines, they can be used to guide the ongoing research toward the more promising classification methods. Findings and experience gained developing classification methods can also feed back into these guidelines.

6. Further work is required to test and develop these guidelines using empirical results. This work will show how optimising a classifier's performance for different evaluation metrics affects the quality of the output price index for a data source, given different properties of the data and may involve both synthetic and collected price data. Therefore, this work will provide a more complete basis using both theory and empirical results for decisions around classification, such as selecting classification methods, optimising hyper parameters and monitoring performance over time.
7. We will then be able to use this to select the most appropriate classification methods for a given data source and ONS item category. This will give us a final recommendation on which method(s) to implement in the pipeline. We are aiming to have these recommendations ready to present to the panels in the beginning of 2020, in line with the milestones set out in APCP-S(19)04 Alternative data sources roadmap.

Recommendations

8. For more information about the classification metrics summarised in this section, please see Annexes A to E of this paper. For readers unfamiliar with classification metrics, it is recommended to read the Annexes first before reading this section.

Binary classification

9. For binary classification we can derive the following general strategies.

    a) Where distributions of our classes are not stable, there is no real advantage to using Fβ – score over balanced accuracy, unless positive class prevalence is stable and low.
    b) With stable and low positive prevalence, Fβ – score is preferred as it works well as a metric to measure correct classification of the positive labels.
    c) With stable and balanced or high prevalence, balanced accuracy is preferred as it works well as a metric to measure correct classification of the positive labels.
    d) In some cases, where it is necessary to emphasise positive or negative errors and where distributions are stable, we might want to consider Fβ – score with balanced and high prevalence data as we can set β to emphasise positive and negative errors. However, balanced accuracy is generally better at evaluating performance for balanced and high prevalence data.
    e) Table 1 summarises the recommendations given particular properties of the data (for example, if the prevalence of the data is stable over time) indicated in different columns. Recommendations for how to measure performance are grouped by row with a brief explanation, for example, what metrics to use, should you consider averaging etc. For each of these it is divided into a what and why section. 'what' gives you options, 'why' provides a brief explanation of the reasoning behind each and recommends a single option if there is more than one.

Table 1: Binary classification summary table

		ARE PREVALENCE AND/OR PRICE DISTRIBUTIONS STABLE OF THE DATA STABLE?			
		Yes	**Only prevalence**	**Only price distribution**	**No**
METRIC / VISUALISATION	What we can use	Fβ– score/PR -space Balanced accuracy /ROC space	Balanced accuracy /ROC space Fβ – score/PR space for low prevalence as alternative to weighted balance accuracy	Fβ – score/PR space Balanced accuracy/ROC space	Balanced accuracy/ROC space
	Why you would use a measure	For low prevalence we use Fβ – score. Otherwise we use balanced accuracy, for high prevalence and balanced data. In some cases, we *might* consider Fβ – score where we want to minimise false positives (emphasise precision) or false negatives (emphasise recall)	Fβ – score is better with low prevalence datasets. Otherwise use balanced accuracy	Fβ – score allows us to choose to minimise false positives (emphasise precision) where price distributions of classes are different or false negatives (emphasise recall) where price distributions of classes are similar However, if we decide to set β = 1 or close to 1 then we may consider balanced accuracy as there is no benefit to Fβ – score, especially with high prevalence or balanced data	Results with balanced accuracy are comparable across datasets so we can monitor performance over time

PREVALENCE WEIGHTING	Should we weight?	Yes	No	No	No
	Why?	We should only weight where we use average Fβ – score as prevalence is unlikely to change, so we can account for different size classes and see class breakdowns. Otherwise it should not be used as balanced accuracy is prevalence insensitive.	We use balanced accuracy is prevalence insensitive and we are not using averaged Fβ – score	Unstable prevalence means these will change if used	Unstable prevalence means these will change if used
AVERAGING	What to use	Macro if classes are unbalanced.	N/A	N/A	N/A
	Why	Class prevalence is stable, so we can use weights to correct for class imbalance when using averaged F-score Balanced accuracy does not require averaging.	Balanced accuracy is macro average of recall for binary problems Not using average F-Score	Not using average F-Score	Balanced accuracy is macro average of recall for binary problems
β	What value to pick?	Set β depending on how often we expect to see false negative or false positive errors.	1	Set β depending on how often we expect to see false negative or false positive errors.	n/a
	Why	As distribution is stable we can tune our metric to reduce the impact of errors	Provides the best compromise as distributions are unstable.	As distribution is stable we can tune our metric to reduce the impact of errors	n/a for balanced accuracy

Multiclass classification

10. For multiclass classification we can derive some general strategies.
    a) We use macro averaging with prevalence weights to correct for unbalanced classes if prevalence is stable as we can see class breakdowns and we know any class imbalance will not change over time, so we can weight appropriately.
    b) Where it is not stable we have to use micro averaging as any prevalence weights we use will not work with future datasets.
    c) Where distributions are stable over time we can inspect these, then set β appropriately to put emphasis on reducing false positive or false negatives – see errors for guidance on how to choose. Therefore, we choose Fβ– score otherwise we will not have the option to do this. One caveat; if you decide the best strategy is to set β = 1, you may also use average recall as a metric, which would be better in situations where true negatives are important. This is summarised in the table below.

Table 2: Multi-class classification summary table

		ARE PREVALENCE AND/OR PRICE DISTRIBUTIONS STABLE?			
		Yes	Only prevalence	Only price distribution	No
METRIC / VISUALISATION	What	Fβ– score/PR -space Or, where β = 1 Av Recall/ROC space	Fβmicro – score /PR -space Av Recall/ROC space	Use unweighted Fβ – score with PR space	Fβ– score/PR -space Av Recall/ROC space
	Why?	We can choose to minimise false positives (emphasise precision) or false negatives (emphasise recall) If we choose to balance these with β = 1 then choose Av Recall If we care about true negatives	If we care about true negatives, we prefer average recall, otherwise use Fβmicro – score	We can choose to minimise false positives (emphasise precision) or false negatives (emphasise recall)	If we care about true negatives, we prefer average recall, otherwise use $F\beta_{micro}$ – score
PREVALENCE WEIGHTING	What	Yes	Yes	No	No
	Why?	Prevalence is unlikely to change, so we can account for different size classes and see class breakdowns	Prevalence is unlikely to change, so we can account for different size classes and see class breakdowns	Unstable prevalence means these will change if used	Unstable prevalence means these will change if used
AVERAGING	What	Macro	Macro	Micro	Micro
	Why	Class prevalence is stable, so we can use weights to correct for class imbalance	Class prevalence is stable, so we can use weights to correct for class imbalance	Micro averaging is not influenced by prevalence and we can't use prevalence-weighted macro as prevalence is not stable.	Micro averaging is not influenced by prevalence and we can't use prevalence-weighted macro as prevalence is not stable.

β	What?	Set β depending on how often we expect to see false negative or false positive errors.	1	Set β depending on how often we expect to see false negative or false positive errors.	1
	Why	As distribution is stable we can tune our metric to reduce the impact of errors	Provides the best compromise if distributions are unstable.	As distribution is stable we can tune our metric to reduce the impact of errors	Provides the best compromise if distributions are unstable.

Future work

1. The guidance presented here can be used to develop classifiers using real price data to maximise the quality of the final price index for that data source and ONS item category. These guidelines will be subject to ongoing development and refinement as the research phase progresses. Directions for further work with regards to classification metrics are listed below.
2. These work streams will allow us to develop and select the most appropriate classification method(s) for a given data source and ONS item category. We will then be able to make recommendations on method(s) to implement in the pipeline as well as how to assess the ongoing performance of the classification module in the pipeline. We are aiming to have these recommendations ready to present to the panels in the beginning of 2020, in line with the milestones set out in APCP-S(19)04 Alternative data sources roadmap.

Quantifying effect of errors on price index

3. Using fully labelled real data and data with synthetic prices and labels, it will be possible to simulate a classifier achieving particular performance levels for different metrics on different datasets. For example, take a dataset classified with a balanced accuracy of 0.9. It would be possible to calculate the index for every possible set of assigned labels that would give a balanced accuracy of 0.9, giving every possible index for this dataset. Variance and standard errors for the index can be calculated, given a defined level and metric and the quality of the indexes can be assessed. This allows us to judge what is the most appropriate metric and level for an acceptable index quality level. This can be repeated for different distributions of price for different classes in a data set to investigate how the quality of the index may be affected when the properties of the price data change over time. Meaning that we can develop proper tests for the data and classification step in the pipeline to ensure the quality of the final price index with each delivery of new data when the system is in continuous production.

Developing cost functions for classification

4. In price data, not all items have equal expenditure. If this item is classified incorrectly, then this may have a substantial impact on the index and/or limit the use of weighted index methods. To account for this, it is possible to use a cost function to determine the cost of classifying each product correctly or incorrectly. A cost function could be constructed from expenditure weights or estimates of expenditure where explicit weights are not available. This can be used to assess classification quality in place of counts of observations classified as true and false positives and negatives.

Guidance on manual checking and re-training of classifiers

5. Classifiers will need to be periodically retrained, depending on properties of the data such as product churn and price and feature stability. Classifier performance on new products needs to be comparable to existing products to avoid biasing the index. This means that classifier performance needs to be manually assessed on a sample of new items over time. If price distributions of True and False labelled items change, this may mean that the classifier needs retraining as the impact of false positives and negatives may have changed. Therefore, there is a requirement to investigate methods of how often performance needs to be assessed and how often classifiers need to be re-trained, every month, 3 months, when certain performance threshold drop below a defined level? Since it is likely that the data is too large to be completely manually checked, a suitable sampling method is required to gain an accurate estimate of classifier performance.

Edward Rowland
Methodology, Office for National Statistics - UK
May, 2019

List of Annexes

Annex D – Visualising classifier performance

Annex A – Classification definitions

Binary (two class) classification

1. This is where we have two classes we want to separate, for example it may be observations we want to include in the index for an ONS item (for example, laptops) and observations that we do not want to include in the index (for example, laptop bags). This is the most common textbook classification task and most classification metrics apply to this. See chapter 2 and 3 in (Flach, 2012) for a more detailed overview of classification and machine learning.

Ground truth

2. In COICOP5 and ONS item definitions we have a defined set of labels we need to classify our observations to, meaning we have a supervised classification problem. We take the ONS item definitions as our ground truth and require a correctly labelled dataset that reflects this. With respect to a chosen category, we define two labels to provide our ground truth.

   Labelled Positive – an observation that should be included in a chosen class (ONS item definition)

   Labelled Negative – an observation that should not be included in a chosen class (ONS item definition)

Classifier judgements

3. Then, we run the data through our classifier and get a set of classification judgements. These are defined in a similar way to labelled observations.

   Classified Positive – an observation that is classified in a chosen class (ONS item definition)

   Classified Negative – an observation that is not classified in a chosen class (ONS item definition)

Correctly and incorrectly classified observations

4. For each observation in our dataset, we compare the classifier's judgement to the label or ground truth for a particular class and we can determine four distinct categories we can place this judgement in.

   True positive – An observation both labelled and classified as positive so is *correctly included in a class*

   True negative – An observation both labelled and classified as negative so is *correctly excluded from a class*

   False positive – An observation labelled negative and classified as positive so is *incorrectly included in a class*

   False negative – An observation labelled positive and classified as negative so is *incorrectly excluded from a class*

5. True positive and true negatives are our two types of correct values and false positive and false negative are our incorrect values. False positive and false negative are type I and type II errors respectively. Typically, we show these values in a confusion matrix below, along with the marginal sum totals of labelled positive and negative observations and classified positive and negative.

Annex A - Table 1: Binary confusion matrix

Label	**Classified**		
	Positive	Negative	**Truth**
Positive	True +ve	False -ve (Type II)	Total labelled +ve
Negative	False +ve (type I)	True -ve	Total labelled -ve
Classified as	Total classified +ve	Total classified -ve	**Total observations**

6. This confusion matrix is what you will see for a binary classification – where there are two groups. For multi-class classification, you can create a much larger matrix, or you can show the above matrix for a single class compared to all others in a one vs rest comparison.

Multiclass classification

7. To measure performance across an entire dataset with multiple classes, if an observation has been placed into the correct class, this is a correct classification, where it has been placed into the incorrect class, it is an error. There is no distinction between True and False positives and True and False negatives as there are as many different types of error as there are incorrect classes. Since most metrics Annex B – Classification metrics use these two types of correct and incorrect classifier judgements, they are not defined in terms of multiple classes. This limits the metrics available for assessing multiclass classification in this confusion matrix (table 2) to accuracy and error. If we wish to use a wider range of metrics, we need to compute them on a class-by-class basis in a one verses rest approach described below. We can still produce a confusion matrix for multiclass classification as below, in this example we show four classes.

Annex A - Table 2: Multiclass confusion matrix

Classified

Label	Class A	Class B	Class C	Class D	**Truth**
Class A	**Correct**	Incorrect	Incorrect	Incorrect	Total labelled Class A
Class B	Incorrect	**Correct**	Incorrect	Incorrect	Total labelled Class B
Class C	Incorrect	Incorrect	**Correct**	Incorrect	Total labelled Class C
Class D	Incorrect	Incorrect	Incorrect	**Correct**	Total labelled Class D
Classified as	Total classified Class A	Total classified Class B	Total classified Class C	Total classified Class D	**Total observations**

One-versus-rest

8. To use a wider range of metrics in multiclass classification, one method is to reduce the complex confusion matrix into multiple smaller, matrices each akin to a binary classification problem. We can do this by comparing each individual class against all other classes in a one-versus-rest matrix. This allows us to see the performance of the classifier for each individual class using the metrics in Annex B but does not give us an idea of overall classifier performance. In the confusion matrix (Table 3) below we take Class A to be the class we are assessing so will be our positive class and class B, C and D we collapse into the negative class.

Annex A - Table 3: One-versus-rest

		Classified	
Label	Positive (Class A)	Negative (Class B, C, D)	**Truth**
Positive (Class A)	True +ve	False -ve (Type II)	Total labelled +ve *(Class A)*
Negative (Class B, C, D)	False +ve (type I)	True -ve	Total labelled -ve (Class B, C, D)
Classified as	Total classified +ve *(Class A)*	Total classified -ve (Class B, C, D)	**Total observations**

Total observations across classes

9. A One-versus-rest confusion matrix only shows classifier performance for a single class. Therefore, to assess performance over all classes requires a matrix for each class in the data which is cumbersome and does not give an overview of classifier performance over all classes. To get metrics for overall performance, we sum the true and false positive and negative classifications for each classes' one-versus-rest confusion matrix together. This gives an overall confusion matrix (table 4). We define each value in our confusion matrix as below. Where C is the observation category (true positive, false positive etc.) and n is the number of different classes.

$$total\ C = \sum_{i=1}^{n} C_{class\ i}$$

This means that with four classes (A,B,C,D) to get the overall true positive value, we first compute the true positives for each class individually in a one-versus-rest comparison, say we get (A = 4, B=7, C=2 and D=5) then we sum these together to get the overall true positive count of 18 for all classes. This is repeated for false positive as well as true and false negatives to get the overall matrix.

Annex A - Table 4: Overall confusion matrix

	Classified		
Label	Positive	Negative	**Truth**
Positive	Total true +ve	Total false -ve (Type II)	Total labelled +ve
Negative	Total false +ve (type I)	Total true -ve	Total labelled -ve
Classified as	Total classified +ve	Total classified -ve	**Total observations**

Prevalence and balanced data

10. Another key concept is balance; do the classes in the data have approximately the same number of observations? This is measured by prevalence, calculated by dividing the number of observations in a class by the total number of observations in all classes. In a balanced dataset, this will approximate 1/n for all classes, for n classes. For example, a balanced binary dataset will give a prevalence of 0.5. If a dataset contains 4 classes, the prevalence will be 0.25 for each class if the dataset is balanced.

$$prevalence = \frac{Labelled\ positive}{Total\ observations}$$

Annex B – Classification metrics

Accuracy

1. Perhaps the most obvious way of measuring classifier performance is accuracy. Accuracy tells us of all the observations in a dataset, how many are classified correctly? This is given as a ratio or a percentage and can be derived from the correct (true positive and true negative classifications) as shown below. Given a balanced dataset, a higher accuracy score means that a classifier is making more correct classifications, though accuracy runs into issues when used with unbalanced data (Akosa, 2017) as with high prevalence data, a high accuracy score does not necessarily mean high performing classifier.

$$Accuracy = \frac{(True\ positive + True\ negative)}{Total\ observations}$$

True positive rate/Recall

2. Recall or the true positive rate, gives the ratio of the observations that have been correctly classified as positive over the total of observations that are labelled positive found by summing the true positive and false negative observations., given below. Or, when an observation should be classified as positive, how often is it classified positive? In Bayesian probability, true positive rate is the conditional probability that a labelled positive value is classified positive. True positive rate gives a measure of how well your classifier is replicating the labelled positive class in its classified positive judgements as a high true positive rate will mean maximising true positives and minimising false negatives. In an equivalent way, it is possible to compute the true negative rate, we just replace the true positive with true negative and false negative with false positive

$$True\ positive\ rate = \frac{True\ positive}{True\ Postive + False\ Negative}$$

False positive rate

3. This gives the ratio of the observations that have been incorrectly classified positive over the total of observations that are labelled negative calculated by summing false positive and true negative observations. Or, when an observation should be classified negative, how often is it classified positive, given below. False positive rate measures how well your classifier avoids mislabelling false values as true. So, a lower false positive means the classifier minimises the number of false positives and maximise the true negatives.

$$False\ positive\ rate = \frac{False\ positives}{False\ positives + True\ Negatives}$$

Precision

4. Precision is like the true positive rate, except it uses the false positive instead of false negative to give the total classified positive in the denominator meaning it gives the true positives over total classified true. Or, for a classified positive observation, how likely is this observation labelled positive? Precision can also be considered the Bayesian posterior probability that an observation is labelled positive, given that it has been classified as positive. So, a classifier with a high precision will have a high number of true positives with a low number of false positives. The main difference between precision and recall/true positive rate is precision is more focused on the certainty if the predictions for the true class are correct, rather than how much of the labelled true class are predicted true by the classifier.

$$Precision = \frac{True\ positive}{True\ positive + False\ positive}$$

F-score

5. F-score is the harmonic mean of recall and precision. It is useful as often, with real-world problems, a classifier needs to make a compromise between reducing false positives at the expense of increasing false negatives, or vice versa. The F-score tells us how well the classifier makes this compromise. It is computed from the harmonic mean as if one of precision or recall scores is 0, F-score would also be zero indicating a poor classifier. F-score is calculated as below.

$$F = 2 * \left(\frac{precision * recall}{precision + recall}\right)$$

Weighted F-Score

6. We can change the relative importance of recall over precision by adding in a weight variable to the equation. This means if a price index can tolerate false positives over false negatives, we should weight recall as more important than precision as a high recall generally means a reduction in false negatives. If the price index can tolerate false negatives over false positives, then we can reduce the importance of recall over precision. It is important not to confuse weighted F-score with F-score from macro averaged precision and recall, which can include specific class weights. So, to avoid confusion a weighted F-score is referred to as Fβ-score where β is the weight. This means an unweighted F-score is equivalent to an F1-score as it is equal to using a weight of 1 in the equation below.

$$F_\beta = (1 + \beta^2) * \left(\frac{precision * recall}{(\beta^2 * precision) + recall}\right)$$

7. One feature of the Fβ-score (including the unweighted F) is that it is not affected by the true negative rate. (Powers, 2015) discusses limitations of the F-score which includes this observation. Though whether this is of any concern depends on the problem a classifier is trying to solve as this makes F-score ideal for needle-in-a-haystack problems. A needle-in-a-haystack type of problem where the number of labelled negative observations is much greater than the number of positive observations i.e. prevalence is low, is an ideal use case for Fβ-score. An example of this is say we want to find all of the different banana products a supermarket sells. Most products are not going to be bananas, therefore the negative class (not a banana) is much greater than the positive class (a banana). This is because performance, with respect to this use-case, rarely depends upon the negative class but how well the classifier does at correctly picking out the positive class. If this is not the case, a different metric would likely provide a better measure of performance.

Balanced accuracy

8. Balanced accuracy is the arithmetic mean of the true positive rate and the true negative rate (positive and negative recall). Unlike Fβ-score it considers the classifiers ability to identify true negatives. It is termed balanced accuracy as it is equivalent to accuracy for a balanced dataset, but it is much better at correctly identifying poor performing classifiers when the data is unbalanced. For example, if we take a naïve classifier that labels all observations as true with a binary classification problem on a dataset with a prevalence of 0.9. The accuracy of this (poor) classifier would be 0.9 as it would correctly label the true observations that make up 90% of the data and only label 10% of the data incorrectly, as just 10% of the observations are false. However, balanced accuracy will be 0.5. As the naïve classifier correctly identifies all the true observations (true positive rate = 1) but incorrectly labels all false observations as true (true negative rate = 0). Following the equation below, the sum of these divided by two gives 0.5. 0.5 is interpreted the same as chance. This means with balance accuracy, we correctly identify our naïve classifier as being bad at solving this problem, but from accuracy we might incorrectly conclude it is good at solving this problem. Hence balance accuracy is preferred. This also provides a useful property of balanced accuracy in that it gives consistent measures across datasets with different prevalence scores.

$$Balanced\ accuracy = \left(\frac{true\ positive\ rate\ (positive\ recall) + true\ negative\ rate\ (negative\ recall)}{2}\right)$$

9. It is worth noting here that balanced accuracy is just the average recall for binary classification. Average recall is defined later in the section on Macro averages. The rest of this document refers to average recall and balanced accuracy interchangeably.

Annex C: Averaging methods for multiclass classification

1. The metrics in Annex B are defined for binary classification, in this section they are expanded for use in multiclass classification using two different methods; micro and macro averages. Micro average metrics are computed from the overall confusion matrix containing the sum total of the true and false negatives and positives for each class as in the Total observations across classes section. Then we compute the metrics, (recall, precision etc.) to assess the classifier. For the macro average, we compute the metric for each individual classes' one-versus rest confusion matrix as in Annex A: One-versus-rest, then take the average of each individual classes' metric. For macro averaging we can also define class weights to give a weighted macro average. Steps to compute both are shown below.

   Micro average
   - Step 1: Compute the one-versus-rest confusion matrices for each class
   - Step 2: Compute the overall confusion matrix by summing the observation types (true and false positives and negatives)
   - Step 3: Calculate the metrics from this overall confusion matrices

   Macro average
   - Step 1: Compute the one-versus-rest confusion matrices for each class
   - Step 2: Compute metrics for each individual classes' one-versus-rest confusion matrix
   - Step 3: Calculate the overall metrics by averaging the metrics for each class, weighing the average by class if required.

2. The difference in computing micro and macro averages is do we sum up the different observation types for all our classes, then compute our metrics. Or, do we compute the metrics for each class and take the average of these metrics.

3. These different methods have different properties. Macro is useful as you can see how each individual class contributes to the overall classifier performance, so it is easy to see if good overall performance is consistent across classes or are there some classes where the classifier is weak. But with unbalance classes, it treats all classes the same so does not consider the size of the class relative to the others. This means that classifying an observation in a smaller class correctly or incorrectly has a greater effect on the metric than classifying and observation in a larger class. Whether this is preferable behaviour depends on the use case. Weighted macro average offers a way to tune this behaviour to the use case if emphasizing smaller classes in not preferable or importance of each class needs to be disassociated with size in other ways.

4. Micro accounts inherently accounts for unbalanced classes by putting all classes together in an overall confusion matrix to calculate metrics. Therefore, the contribution to the overall micro averaged metric of any individual observation's correct or incorrect classification is the same as any other. Meaning that micro averaging is not sensitive to class prevalence within the data and gives it the useful property of providing comparable metrics across different datasets.

Micro average

5. In micro averaging, recall/true positive rate and precision are defined as with binary classification, except the total observations from each class's one-verses-rest comparison for each category are used. Therefore, the first order metrics become those defined below.

$$True\ positive\ rate_{micro} = \frac{total\ true\ positive}{total\ true\ postive + total\ false\ negative}$$

$$precision_{micro} = \frac{total\ true\ positive}{total\ true\ postive + total\ false\ positive}$$

6. The second order metrics are defined as follows. Recall and true positive rate are equivalent, and so is true negative when labelled and classified positive and negative categories are inverted.

$$Balanced\ accuracy_{micro} = \left(\frac{true\ positive\ rate_{micro} + true\ negative\ rate_{micro}}{2}\right)$$

$$F_{\beta\ micro} = (1+\beta^2) * \left(\frac{precision_{micro} * recall_{micro}}{(\beta^2 * precision_{micro}) + recall_{micro}}\right)$$

Macro average

7. Instead of taking the categorical observation counts for each class from the one-verses-rest comparisons, each of the first order metrics are calculated on a class-by-class basis, then the arithmetic average of these are taken. So, precision and recall are defined as follows, with the macro F-score then defined by these in turn.

$$recall_{macro} = \frac{\sum_{i=1}^{n} recall_{class\ i}}{n}$$

$$precision_{macro} = \frac{\sum_{i=1}^{n} precision_{class\ i}}{n}$$

$$F_{\beta\ macro} = (1+\beta^2) * \left(\frac{precision_{macro} * recall_{macro}}{(\beta^2 * precision_{macro}) + recall_{macro}}\right)$$

8. Notice that Balanced accuracy is an expression of average recall for two (binary) classes, therefore there is no need to define macro balanced accuracy. Consequently, balanced accuracy and average recall are used interchangeably.

Weighted macro averaging

9. As first order metrics are given by class, we can define class weights to make sure our classifier performs best on the most important classes. A common weighting method is relative class prevalence, so the classifier performs best on the most frequently occurring classes, described here. Definitions are as follows; x_{class} as the number of observations

labelled as in that class, $\bar{x}$ as the mean number of observations in each class to compute a weight, w_{class} by the following.

$$w_{class} = \frac{x_{class\ i}}{\bar{x}}$$

10. Then compute the weighted average metrics as the following.

$$recall_{weighted\ macro} = \frac{\sum_{i=1}^{n} w_{class} * recall_{class\ i}}{n}$$

$$precision_{weighted\ macro} = \frac{\sum_{i=1}^{n} w_{class} * precision_{class\ i}}{n}$$

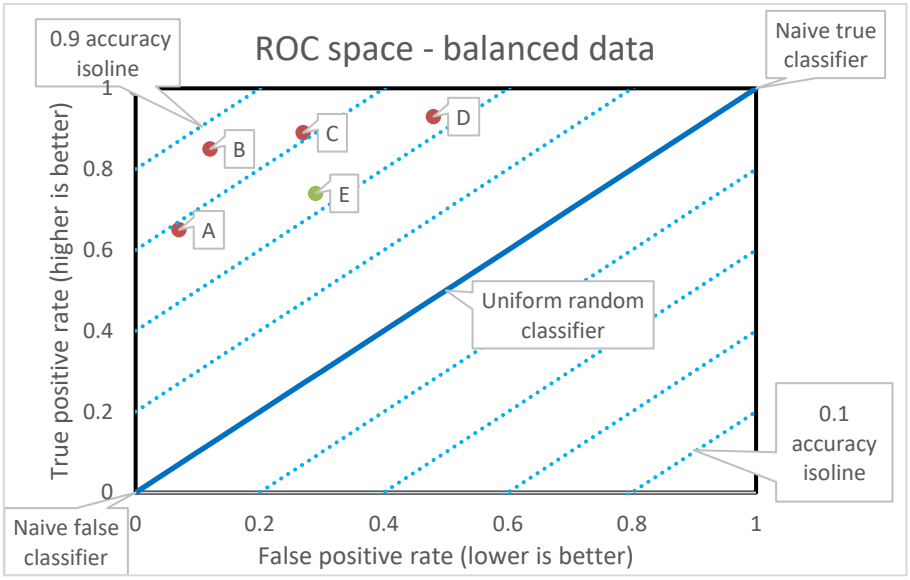
$$F_{\beta\ weighted\ macro} = (1+\beta^2) * \left(\frac{precision_{weighted\ macro} * recall_{weighted\ macro}}{(\beta^2 * precision_{weighted\ macro}) + recall_{weighted\ macro}} \right)$$

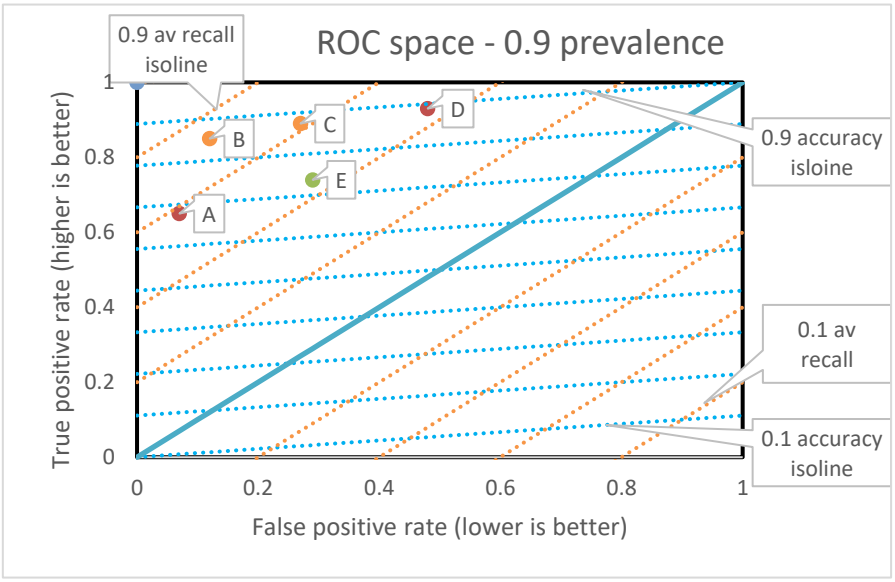
Annex D – Visualising classifier performance

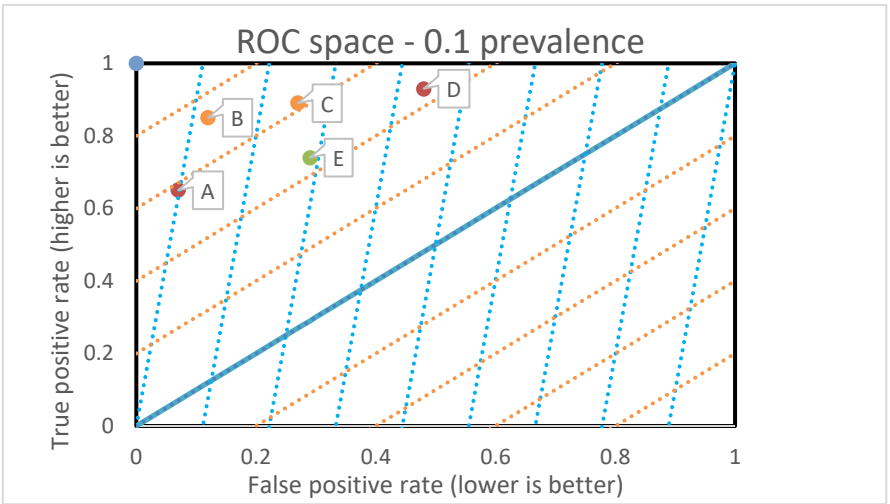
Receiver operating characteristics space

1. We can take two metrics, the true positive rate (Recall) which we want to maximise and false positive rate that we want to minimise to zero and plot these as below in figure 1. This we call receiver operating characteristic (ROC) space. In ROC space, the closer a classifier is to the top left means the closer its performance is to a true positive rate of 1 and a false positive rate of 0 which would indicate a perfect result. To give an example, several example classifiers are plotted in ROC space, labelled A- E. In addition, we label where naïve (where all observations are given the same label) and uniform random classifiers (where for binary classification half the observations are given one label and half the other, allocated at random). These give a minimum benchmark performance, indicated by the solid blue line in figure 1. In our example classifiers, A gives the best false positive rate, while D gives the best true positive rate. B and C offer good compromises between these so could still be useful for us. E is dominated by the other classifiers as it has inferior performance compared to the others on either of these metrics. Therefore, E is ignored as when a classifier is dominated, it is never going to give us the best compromise no matter the use-case.
2. To help further in assessing classification quality, we can use isolines on our plot. These show the points on the plot where the classifier achieves a particular value for a metric. On figure 1 below, we show accuracy isolines at 0.1 intervals. The isoline shows all points in ROC space where a classifier would have a certain accuracy. Isolines for other metrics can be shown also. However, where we have unbalanced data the accuracy isolines shift, so we use balanced accuracy (average recall – the orange lines) instead as this is invariant to class prevalence and is equivalent to accuracy for balanced classes. We can also see that a naïve classifier is as accurate if not more accurate than our 'smart' classifiers on unbalanced data. This is because with a prevalence of 0.9 the naïve true classifier scores 0.9 accuracy as it correctly labels 90% of the observations as 90% are true observations. With 0.1 prevalence, the naïve false classifier performs similarly well for the same reason. Therefore, balanced accuracy is preferred when using unbalanced data and demonstrates how ROC space with appropriate isolines provides a useful tool to visualise classifier performance.

Annex D - Figure 1: ROC space plots





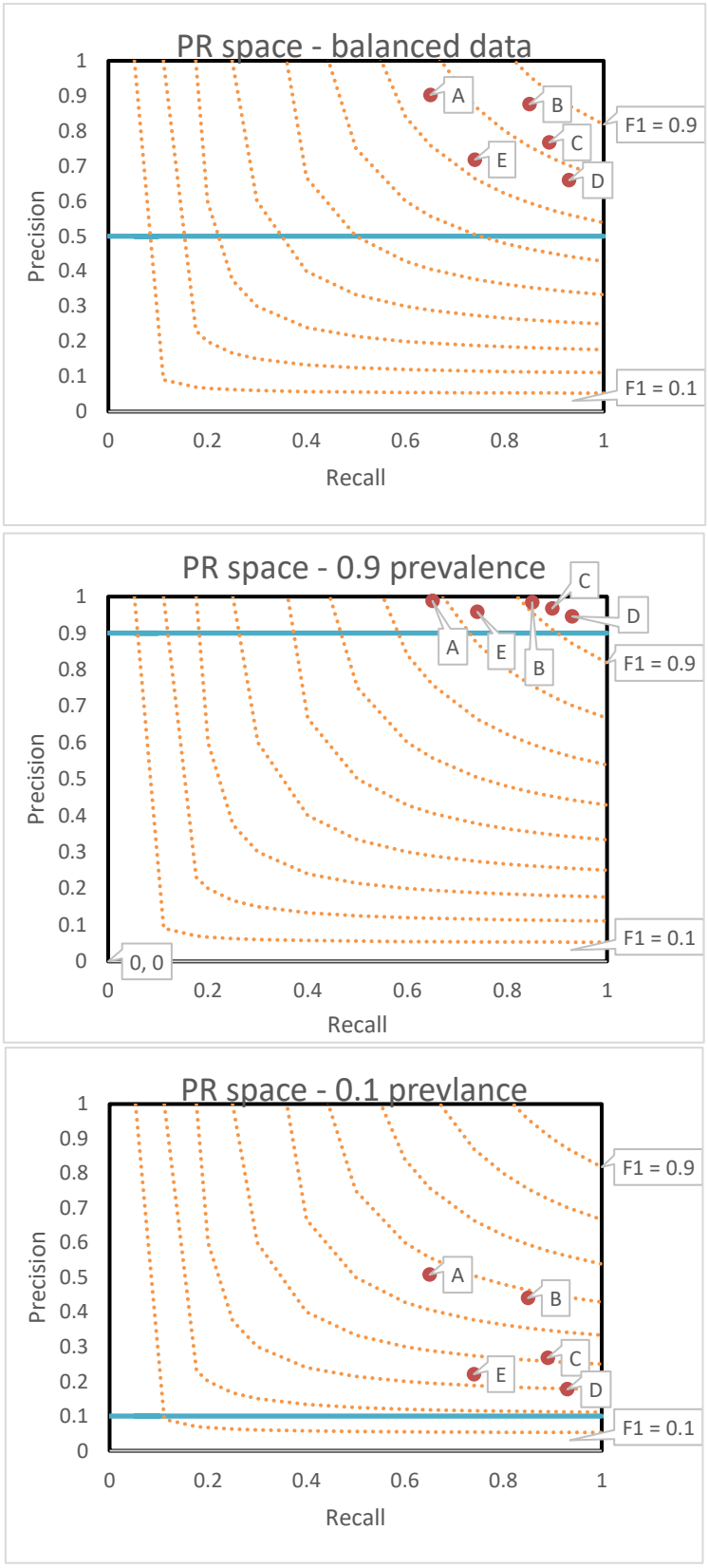


3. We can see the classifier B gives the best compromise between true and false positive rates as it has the highest balanced accuracy/average recall; the operative word being compromise. We should not discount the other classifiers as if we are sensitive to false positive rates, we may want to choose A, if we are tolerant of them we may want to choose C or D, it depends on use case. E however is poor compared to the other classifiers, in no situation will it offer the best compromise depending on if we want to minimise false positives or negatives. This is termed dominance, so we say E is dominated by the other classifiers.

Precision-recall space

4. Precision and recall are two other metrics that can be used to assess a classifier. While it is possible to have perfect precision across all levels of recall, this rarely happens in real world use cases, so a compromise between these is made. Precision-Recall (PR) space can be used to visualise this compromise in an equivalent way to the ROC space plot with recall on one axis and precision on the other. As we aim to maximise both precision and recall, the ideal classifier is at the top right. A useful property of PR space is that Fβ-score is the weighted harmonic mean of both precision and recall, so we can easily plot isolines for our Fβ-score to get a direct measure of how well the classifier is making the compromise. The precision of the uniform random and naïve classifiers is the same, equal to the prevalence, but the recall is different, we take the baseline performance of precision as equal to the prevalence of the labelled positive class across all recall values. This is represented as a horizontal line across PR space, parallel to the recall axis, intersecting the precision axis at our labelled positive prevalence value.
5. Figure X plots the PR space using the same classifiers with the same true positive (recall) and false positive rates as with ROC in Figure 2. No classifier is below the blue line that indicates when performance is extremely poor. Again, B gives the best compromise between precision and recall, indicated by its proximity to the 0.9 F1-score isoline. C and D give us options if we can tolerate a lack of precision, and A improves precision slightly on B at a loss of recall. Where the prevalence of the data changes, so does the baseline performance. For majority positive (0.9 prevalence), the classifiers are compacted on the vertical axis making it difficult to make out which offers the highest precision. With majority negative data (0.1 prevalence) the classifiers are much more spread out against the vertical axis. Intuitively, this demonstrates it is harder to score more highly on precision when the dataset consists of labelled negative observations and more clearly shows the difference in performance for our classifiers. Therefore, PR space is an effective tool to use with low prevalence data, but not with high prevalence.

Annex D - Figure 2: PR space plots



Annex E – Assessing impact of classification errors

1. It is extremely unlikely that we are going to have a classifier that makes no errors. Therefore, we have to take the view to build a classifier which minimises the impact of errors on a price index. We can examine what the potential impact of false positive and false negative errors might be in different situations to determine how best to build our classifier. This will depend on the price level and distribution of errors compared to the classified and labelled positive and negative observations, which we can investigate below. The exact pricing variable that will need to be investigated will depend upon the index methodology and so could be price relatives, unit price or others. In the examples in this section, price is referred to and should be interpreted as a catch all term for the relevant pricing variable for the index method being used.

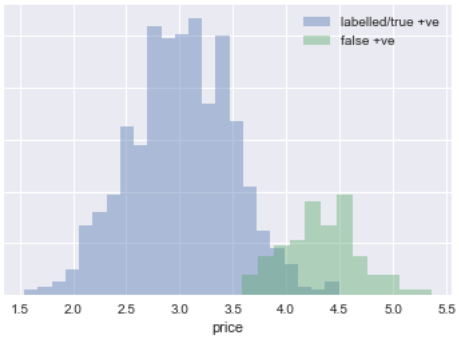
False positive errors

2. False positive errors are observations that should not be including in a price index but have been incorrectly classified so that they are included. What needs to be checked is if their inclusion biases the index in some way. By comparing the price distribution of the labelled positive and classified positive distributions with the false positive errors to see if the distribution or the levels are different. If they are not, the likely impact of false positives will be minimal, shown in figure 1. If the level and/or distribution is different, then this will likely bias the index in some way.

Annex E - Figure 1: Where false positives impact on the index is (probably) minimal

3. Figure 2, shows an example where the level of false positive errors are biasing the distribution of the classified positive observations so they are different to the labelled true distribution. This means that false positives may introduce bias into the index.

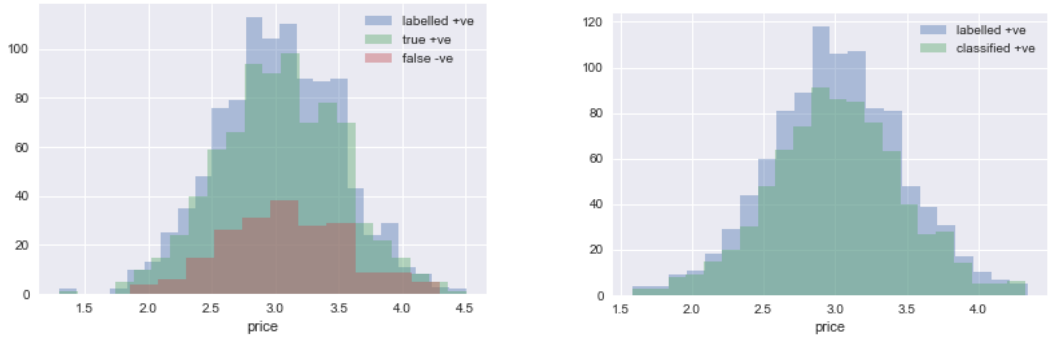
Annex E - Figure 2: Where false positives level is (probably) affecting the index



False negative errors

4. False negative error is when the classifier incorrectly excludes an observation from the index that should be included. If false negatives are randomly distributed in price, then their exclusion will be unlikely to skew the distribution of classified true observations from the labelled true observations distribution. If they are not, then their exclusion might skew the classified true distribution away from the labelled true distribution.
5. If the distribution of false negatives is similar to the labelled true and the true positive observations i.e. false negatives are a random sample of the labelled true observations. Then we do not have to worry about false negatives too much. They will have negligible effect in biasing the classified true observations as these will still have a similar distribution to the labelled true observations, assuming no false positives – shown in figure 3.

Annex E - Figure 3: Where false negative impact on the index is (probably) minimal



6. If false negative distributions are different to the labelled true distributions then we do need to be concerned about the effect of false negatives on the index as this may cause this distribution of classified true to differ from the labelled true distributions which will bias the index, again assuming no false positives.

Annex E - Figure 2: Where false negative distribution is (probably) affecting the index

Multiclass classification

7. Multiclass classification is more complex. One approach is to perform similar checks for each class in one-verses-rest comparisons, but the likelihood is that not every class will be affected by false negatives or false positives in the same way. If they are we can follow the same as binary classification, otherwise the requirement will be to balance performance for the different types of errors and choosing a metric accordingly.

Annex F - Data representation and stability over time

1. We must consider how representative our training data is of the real-world, including the stability of the data properties that might affect the price index. Domain knowledge and past data can be used to examine how the class has changed over time and get an idea if prevalence or price distribution of item classes are stable over time. Unstable prevalence and/or price distributions means predicting the impact of different errors or how much different classes should contribute to classification quality is difficult if these properties change. If these properties are liable to change over time, then the classifier needs to give a

good balance when measuring performance across classes and provide a good trade-off between positive and negative errors, so the method is robust to future changes in the data. Despite meaning the classifier and index might not be optimised to the particular prevalences and price distributions of the training data. This means choosing metrics that are insensitive to class prevalence and do not emphasise one type of error or the other. One option is average recall, another is $F1_{micro}$ – score. Both are insensitive to class prevalence as well so are a good option where prevalence is likely to change over time.

Annex G - References

Akosa, J. S. (2017). Predictive Accuracy : A Misleading Performance Measure for Highly Imbalanced Data. SAS® GLOBAL FORUM 2019. Dallas, TX**.**

Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. *ICML '06 Proceedings of the 23rd international conference on Machine learning* (pp. 233-240). Pittsburgh, Pennsylvania: ACM New York.

Flach, P. (2012). *Machine Learning: The Art and Science of Algorithms that Make Sense of Data.* UK: Cambridge University Press.

López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 113-141.

Powers, D. M. (2015). What the F-measure doesn't measure: Features, Flaws, Fallacies and Fixes. *ArXiv*.

Saito, T., & Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*.