IDENTIFYING INCORRECT ANNOTATIONS IN MULTI-LABEL CLASSIFICATION DATA

Aditya Thyagarajan, Elías Snorrason, Curtis Northcutt, Jonas Mueller
Cleanlab
{aditya,elias,curtis,jonas}@cleanlab.ai

ABSTRACT

In multi-label classification, each example in a dataset may be annotated as belonging to one or more classes (or none of the classes). Example applications include image (or document) tagging where each possible tag either applies to a particular image (or document) or not. With many possible classes to consider, data annotators are likely to make errors when labeling such data in practice. Here we consider algorithms for finding mislabeled examples in multi-label classification datasets. We propose an extension of the Confident Learning framework to this setting, as well as a label quality score that ranks examples with label errors much higher than those which are correctly labeled. Both approaches can utilize *any* trained classifier. Here we demonstrate that our methodology[1] empirically outperforms other methods for label error detection. Applying our approach to CelebA, we estimate over $30,000$ images in this dataset are incorrectly tagged.

1 Introduction

Many real-world datasets contain label errors, which should be identified and fixed to train the best models for supervised learning (Natarajan et al., 2013; Lee et al., 2018; Song et al., 2022; Huang et al., 2019; Northcutt et al., 2021a). Label errors are particularly likely in tasks where annotators must make many choices for each individual example in the dataset, such as structured prediction (Reiss et al., 2020). Finding the mislabeled examples can be done much more efficiently assisted by algorithms (Kuan & Mueller, 2022). Confident Learning algorithms offer a principled approach to detecting mislabeled examples in multi-class classification datasets (Northcutt et al., 2021b). Alternatively one may rank examples by their mislabeling likelihood, as estimated via a *label quality score* (Kuan & Mueller, 2022), which enables efficient review of the most suspicious labels.

Detecting label errors has been studied for *multi-class* classification (Brodley & Friedl, 1999; Müller & Markert, 2019; Northcutt et al., 2021b) and other tasks like token classification (Wang & Mueller, 2022) or image segmentation (Rottmann & Reese, 2022). However little research exists on **label error detection** in *multi-label* classification datasets. In multi-label classification, each example can belong to one or more of K classes or none of them. It is equivalent to solving K binary classification problems, in a one-vs-rest fashion, where each class either applies to a particular example or not. In contrast, the K classes in multi-class classification are mutually exclusive. While the term *one-vs-rest* commonly refers to a strategy of training independent binary classifiers, we do not advocate for such an approach that ignores inter-class correlations. We use *one-vs-rest* to refer to a general viewpoint of multi-label classification, where each class either applies to each example or not.

From the one-vs-rest perspective, annotators for multi-label classification must consider K questions when labeling each example, thus increasing the potential for annotation errors. Here we consider two approaches for label error detection, called *Flagger* and *Scorer* methods by Klie et al. (2022). Flaggers estimate *which* examples are mislabeled, and Scorers estimate a label quality score that should be monotonically related to the likelihood it is correctly labeled. These approaches can help estimate the number of mislabeled examples and prioritize which ones should be re-examined under a limited label verification budget, respectively. Our algorithms for label error detection are entirely model-agnostic and can utilize any multi-label classifier, regardless of how it was trained.

[1]Code to run our method: `https://github.com/cleanlab/cleanlab`
Reproduce our benchmarks: `https://github.com/cleanlab/multilabel-error-detection-benchmarks`

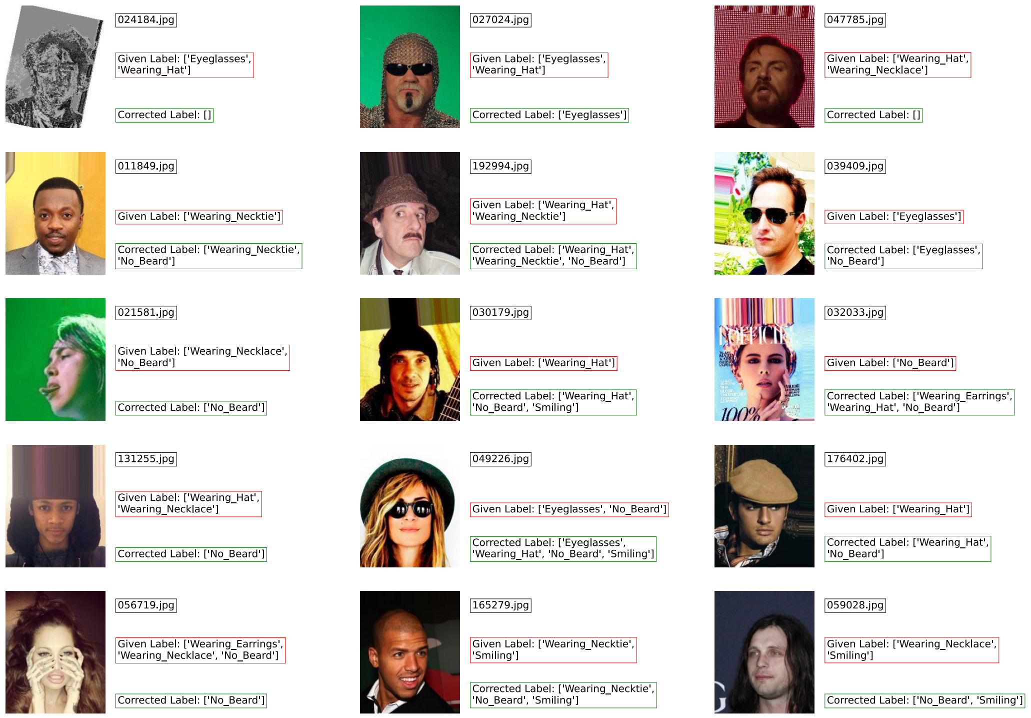

024184.jpg
Given Label: ['Eyeglasses', 'Wearing_Hat']
Corrected Label: []
027024.jpg
Given Label: ['Eyeglasses', 'Wearing_Hat']
Corrected Label: ['Eyeglasses']
047785.jpg
Given Label: ['Wearing_Hat', 'Wearing_Necklace']
Corrected Label: []
011849.jpg
Given Label: ['Wearing_Necktie']
Corrected Label: ['Wearing_Necktie', 'No_Beard']
192994.jpg
Given Label: ['Wearing_Hat', 'Wearing_Necktie']
Corrected Label: ['Wearing_Hat', 'Wearing_Necktie', 'No_Beard']
039409.jpg
Given Label: ['Eyeglasses']
Corrected Label: ['Eyeglasses', 'No_Beard']
021581.jpg
Given Label: ['Wearing_Necklace', 'No_Beard']
Corrected Label: ['No_Beard']
030179.jpg
Given Label: ['Wearing_Hat']
Corrected Label: ['Wearing_Hat', 'No_Beard', 'Smiling']
032033.jpg
Given Label: ['No_Beard']
Corrected Label: ['Wearing_Earrings', 'Wearing_Hat', 'No_Beard']
131255.jpg
Given Label: ['Wearing_Hat', 'Wearing_Necklace']
Corrected Label: ['No_Beard']
049226.jpg
Given Label: ['Eyeglasses', 'No_Beard']
Corrected Label: ['Eyeglasses', 'Wearing_Hat', 'No_Beard', 'Smiling']
176402.jpg
Given Label: ['Wearing_Hat']
Corrected Label: ['Wearing_Hat', 'No_Beard']
056719.jpg
Given Label: ['Wearing_Earrings', 'Wearing_Necklace', 'No_Beard']
Corrected Label: ['No_Beard']
165279.jpg
Given Label: ['Wearing_Necktie', 'Smiling']
Corrected Label: ['Wearing_Necktie', 'No_Beard', 'Smiling']
059028.jpg
Given Label: ['Wearing_Necklace', 'Smiling']
Corrected Label: ['No_Beard', 'Smiling']

Figure 1: 15 of the top-50 scoring images in the CelebA dataset based our proposed EMA label-quality-score, which are also flagged by our Confident Learning extension. Here we choose not to show many additional label errors related to incorrect annotation of the `no_beard` tag amongst these top-50 images, in order to highlight diverse label errors in this dataset automatically detected by our methodology. The dataset has both extraneously added tags as well as many missing tags.

2 Methods

We assume a multi-label classifier has already been trained used to obtain predicted class probabilities $p_i \in \mathbb{R}^K$ for each example i in our dataset. To remain robust against overfitting, these predictions should be out-of-sample, meaning p_i is produced by a copy of the model that has never been trained on the ith example. This can be achieved through cross-validation. The K probability values in any p_i need not sum to 1 since the classes in multi-label classification are not mutually exclusive. A major strength of our proposed methods is how straightforward they are. The sole inputs required are these predicted class probabilities and the given labels for the dataset. No access to the original features or any form of nonstandard modeling is required. Given more accurate predicted class probabilities, our label error detection methods can immediately better identify annotation errors.

Throughout it is useful to adopt a one-vs-rest perspective, considering for each datapoint the binary task of whether each label applies or not. For each of the K classes, form a binary label b_i^k whether class k applies to example i or not, according to the given multi-label annotation. We extract the corresponding kth entry from the predicted class probabilities p_i (denoted as p_i^k) as an estimate of the probability that $b_i^k = 1$. For each class k, Confident Learning is independently applied to these binary labels and predicted binary probabilities p_i^k to flag examples with estimated wrong binary label b_i^k annotations. By combining the *union* of these binary errors detected over all classes $k = 1, ..., K$ for each example, we can report the subset of examples in the multi-label classification dataset estimated to contain an error in their annotation. This approach entails a straightforward extension of Confident Learning to multi-label settings, detailed further in Appendix A.

2.1 Scoring Label Quality in Multi-Labeled Data

A limited budget may prohibit exhaustive verification of candidate examples flagged by our Confident Learning extension. To prioritize examples, we propose numeric *label quality scores* to rank them by our estimated confidence that each one is correctly labeled (Kuan & Mueller, 2022). Under the one-vs-rest perspective, each example has K binary labels b_i^k, each with an associated label quality score s_i^k that estimates how likely a particular b_i^k is incorrect. Verifying b_i^k for an example i and class k requires effort to understand x_i, so it is better to simultaneously verify the other labels for this example than to review a different example.

Hence a key focus of this paper is to estimate **a single label quality score s_i^* for each multi-label example in a dataset**, where s_i^* should be smaller for those examples x_i with any incorrect annotations. Our primary aim is to ensure these label quality scores achieve high precision/recall for detecting examples with any annotation error. Additionally, the lowest s_i^* should correspond to severely mislabeled examples with multiple incorrect labels, as these can have more damaging effects than single incorrect labels (see Table 2), and thus it is more critical to spot them.

Favoring a straightforward approach, we propose the following to compute s_i^*: compute a separate label quality score s_i^k for each binary label b_i^k corresponding to a particular class k, then pooling $s_i^1, ..., s_i^K$ into a single overall score for example i. We focus on *self-confidence* as a binary label quality score (Kuan & Mueller, 2022; Northcutt et al., 2021b; Klie et al., 2022), which is the classifier-estimated likelihood of the given label:

$$s_i^k = b_i^k \cdot p_i^k + (1 - b_i^k) \cdot (1 - p_i^k) \tag{1}$$

Pooling Scores. One can consider several methods to pool the K scores $\mathbf{s}_i \equiv (s_i^1, ..., s_i^K)$ into a single score, s_i^*, for the ith example in a dataset. These methods include taking the lowest, highest, mean, or median score as the aggregated score. The minimum score can be viewed as a bound on the probability of at least one $b_i^1, ..., b_i^K$ being incorrect leading to better precision/recall. It does not necessarily prioritize examples for which many of $b_i^1, ..., b_i^K$ are incorrect. The mean and median scores should better reflect all $s_i^1, ..., s_i^K$, which can be helpful to prioritize examples with many errors in $b_i^1, ..., b_i^K$, but can be more sensitive to nuisance variation (e.g. due to estimation error) in the $s_i^1, ..., s_i^K$ for correctly annotated $b_i^1, ..., b_i^K$.

To achieve a score that accounts for all classes' scores but emphasizes the score of the most suspect class annotation, we propose pooling $s_i^1, ..., s_i^K$ via an **exponential moving average** (**EMA**). Letting $\check{s}_i^1 \geqslant \cdots \geqslant \check{s}_i^K$ denote the sorted binary label quality scores of an example i, we run through them in descending order and report their exponential moving average as an overall label quality score:

$$s_i^* = S_i^K, \text{ where } S_i^t = \begin{cases} \check{s}_i^1 & \text{for } t = 1 \\ \alpha \cdot \check{s}_i^t + (1 - \alpha) \cdot S_i^{t-1} & \text{for } t > 1 \end{cases} \tag{2}$$

Here $0 < \alpha < 1$ is a forgetting-factor constant that is fixed a priori. It defines the relative importance of the s_i^k based on where they occur in the sorted ordering. We propose to use $\alpha = 0.8$, such that s_i^* is most heavily influenced by the binary label b_i^k with the smallest s_i^k value, while still being directly affected by all $s_i^1, ..., s_i^K$ values. This aligns with our primary aim to ensure s_i^* achieves the highest precision/recall for detecting examples with any error in $b_i^1, ..., b_i^K$ while also favoring examples for which many of these binary labels are incorrect. Figures 9-12 show the effect of different α values.

Additional pooling methods evaluated in our benchmark study are presented in Appendix B.

3 Experiments

For evaluation, we consider two groups of multi-label classification datasets (one *large*, one *small*) that contain examples with multiple noisy labels and bag of words features. Each group has ten synthetic datasets, with differences in sample size, feature- and class counts, expected number of classes per example, and overall degree of label noise. We know the ground truth labels and can evaluate the label error detection performance. See Appendix C for more details.

To produce predicted class probabilities for finding label errors, we train two multi-label classifiers, logistic regression and random forest, on each dataset using one-vs-rest 5-fold cross-validation. The random forest model tends to be more accurate than the logistic regression model for the *Small* datasets, but its predicted class probabilities vary less smoothly across different feature values (Table

2). The differences in between these models' accuracies for the *Large* datasets is much lower, where classes are more likely to be linearly separable.

By comparing ground truth labels with noisy given labels, we evaluate label quality scores for each example (estimated from the given labels) via various precision/recall metrics. Here we report **AP @ T** and **Average 2-Precision @ T**, with additional evaluation metrics in the Appendix. Higher values correspond to a better performing method. **AP @ T** is the Average Precision of the label quality score in detecting mislabeled examples among the bottom-T scoring examples. **Average 2-Precision @ T** measures how well scores prioritize examples with at least two incorrect binary labels, among the bottom-T scoring examples.

Figures 2, 3 and 6 to 8 show that across all metrics, EMA performs well relative to the other overall label quality scoring approaches. Weighted cumulative average pooling performs similarly to EMA, but is not as effective. Pooling approaches with minimum-like behavior are good at detecting examples for which any tag is incorrectly chosen, while other approaches like cumulative average pooling are better at detecting severely mislabeled examples for which many tags are incorrect.

These results validate that one can better detect examples with any sort of label error via methods like min-pooling which are not influenced by most of the per-class label quality scores for a particular example. However, accounting for more per-class scores beyond just the minimum score helps better detect severely mislabeled with many errors in more than one of their per-class annotations. Of the alternative methods, those most similar to EMA (like our weighted cumulative average and softmin pooling) perform the best. Using a typical multi-label classification training loss (equivalent to log transform pooling) is not as effective for overall label quality scoring as EMA and similar approaches.

Figures 9 to 12 reveal the effect of different α values in our EMA label quality scoring method. As α grows toward 1, the overall EMA score s_i^* becomes increasingly dominated by the smallest of the per-class scores $s_i^1, ..., s_i^K$. Our benchmarks reveal this generally leads to better detection of examples where *any* of the per-class annotations $b_i^1, ..., b_i^K$ are incorrect (Figure 9) but worse detection of the severely mislabeled examples for which many of $b_i^1, ..., b_i^K$ are incorrect (Figures 10-12). $\alpha = 0.8$ appears to effectively address both objectives. Since the weight of the k-th smallest per-class score $\hat{s}_i^k$ is $\alpha(1-\alpha)^{k-1}$ in the final moving average, with $\alpha = 0.8$, the overall EMA score s_i^* is 3.2% determined by the 3rd lowest per-class score $\hat{s}_i^3$ and only 0.64% determined by the 4th lowest per-class score $\hat{s}_i^4$.

3.1 Finding label errors in the CelebA image tagging dataset

To demonstrate our approach in an image tagging application, we consider the CelebA dataset (Liu et al., 2015). CelebA is a face attributes dataset depicting images of celebrities labeled with various attributes. Here we consider only the following subset of the original CelebA tags (and the subset of 188,000 images annotated with at least one of the tags under consideration): `Wearing_Hat`, `Wearing_Necklace`, `Wearing_Necktie`, `Eyeglasses`, `No_Beard`, and `Smiling`.

We train a neural network for multi-label classification by fine-tuning a pretrained network backbone (efficientnet (Tan & Le, 2019) implemented in the TIMM library (Wightman, 2019)) with a K-dimensional linear output layer added for our prediction task. Each output node uses an independent sigmoid activation rather than a softmax activation which would constrain the predicted class probabilities to be mutually exclusive. Rather than training this classifier in a one-vs-rest fashion, we fine-tune the model using Adam (Kingma & Ba, 2015) to jointly optimize a binary cross-entropy loss at the output node for each class, such that the classifier can learn to model correlations between tags. We produce held-out predictions for each image in the dataset via 4-fold cross-validation.

Running our multi-label extension of Confident Learning and sorting the flagged examples based on our EMA label quality score reveals that CelebA contains many mislabeled examples (Figure 1). There are both extraneously added tags as well as many missing tags in the dataset. In particular, CelebA contains a `No_Beard` tag that should actually apply to a large fraction of the images in the dataset, yet is often not present in the given label. The annotations may contain other inconsistencies reflecting likely annotator confusion regarding when tags like `Wearing_Hat`, `No_Beard`, or `Eyeglasses` should apply (Figure 4).

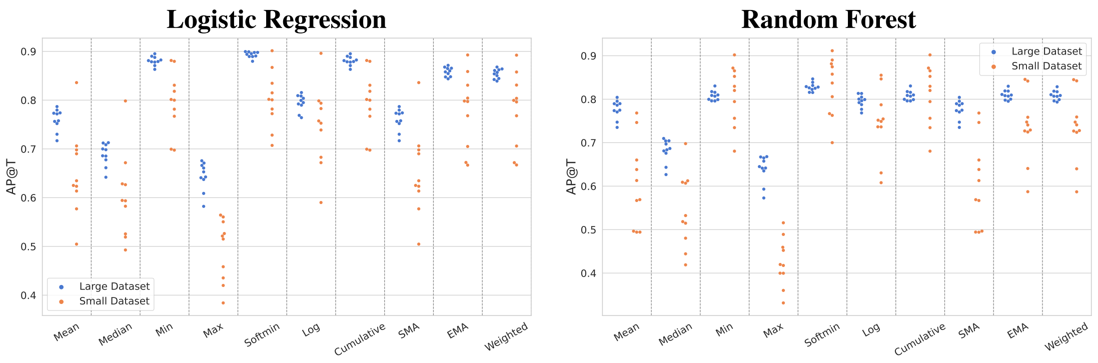


Figure 2: Average Precision @ T achieved by various overall label quality scores s_i^* for each dataset (shown as dot), where T is the number of mislabeled examples in the dataset. We show results based on predicted class probabilities from both a Logistic Regression model and a Random Forest model.

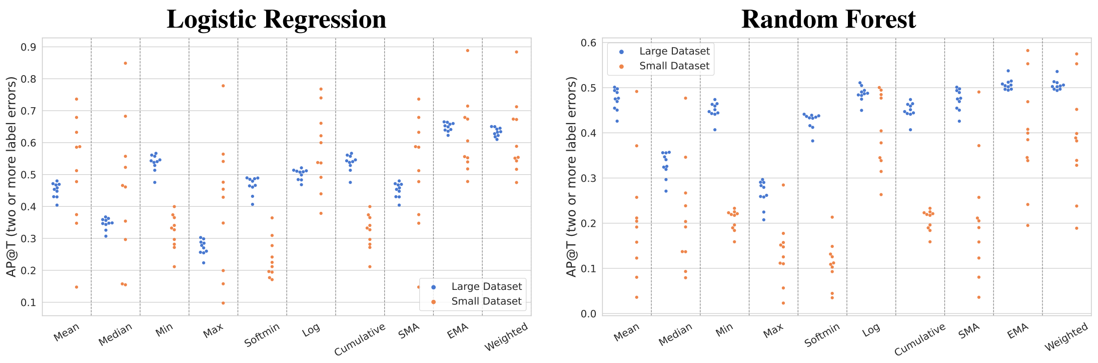


Figure 3: Average 2-Precision @ T achieved by various overall label quality scores s_i^* for each dataset, where T is the number of mislabeled examples in the dataset. We show results based on predicted class probabilities from both a Logistic Regression model and a Random Forest model.

We conducted a small expert validation experiment and found that among 100 randomly chosen images from the dataset 15 were mislabeled – this suggests the the CelebA dataset may contain approximately 15% (30,000) mislabeled images! Among the top-100 images ranked according to our EMA label quality score s_i^*, we identify 67 mislabeled images. Thus the Lift @ 100 for detecting annotation errors via our approach is around 4.5, i.e. **mislabeled images are 4.5 times more prevalent among the set prioritized by our label quality score compared to the overall dataset**.

4 Discussion

This paper introduced model-agnostic methods to identify which examples are mislabeled in multi-label classification datasets and score the confidence of these estimates. A key question to extend these capabilities from multi-class classification to multi-label settings is how to define a single overall label quality score per example. Our proposed EMA score effectively detects examples whose annotated labels contain errors, prioritizing severely mislabeled examples whose label contains many errors.

Beyond images, the approach presented here can be used to easily find such annotation errors in any type of multi-label classification dataset, as long as a reasonable classifier can be fit to the data. Following the spirit of *data-centric AI* (Ng et al., 2021), our model-agnostic methodology can be used with any existing model to find and fix errors in its training set, in order to subsequently train a better model. As more accurate models are invented, our proposed methods can use their improvements to more accurately detect label errors in the same data without modification.

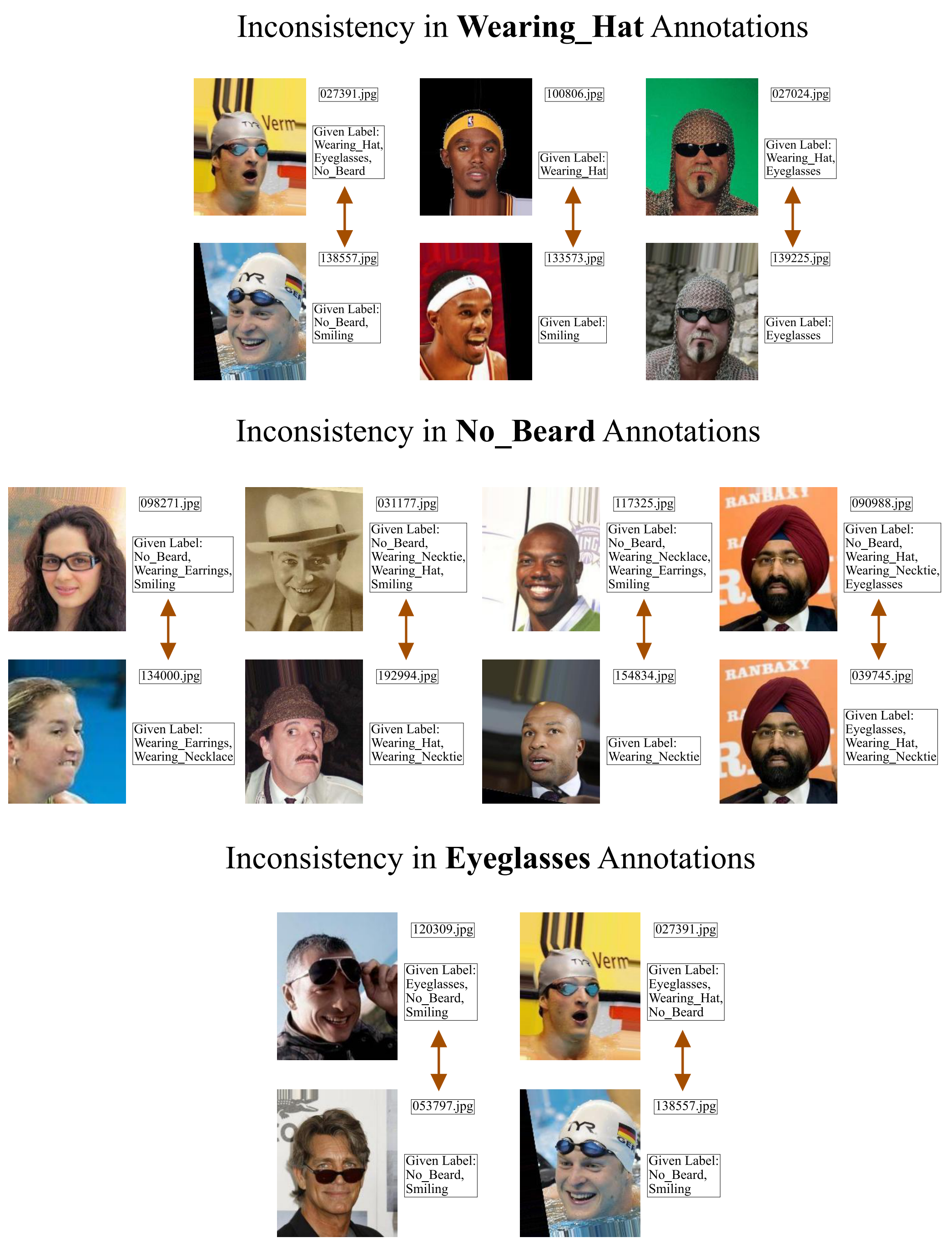

Inconsistency in **Wearing_Hat** Annotations
027391.jpg
Given Label: Wearing_Hat, Eyeglasses, No_Beard
138557.jpg
Given Label: No_Beard, Smiling
100806.jpg
Given Label: Wearing_Hat
133573.jpg
Given Label: Smiling
027024.jpg
Given Label: Wearing_Hat, Eyeglasses
139225.jpg
Given Label: Eyeglasses
Inconsistency in **No_Beard** Annotations
098271.jpg
Given Label: No_Beard, Wearing_Earrings, Smiling
134000.jpg
Given Label: Wearing_Earrings, Wearing_Necklace
031177.jpg
Given Label: No_Beard, Wearing_Necktie, Wearing_Hat, Smiling
192994.jpg
Given Label: Wearing_Hat, Wearing_Necktie
117325.jpg
Given Label: No_Beard, Wearing_Necklace, Wearing_Earrings, Smiling
154834.jpg
Given Label: Wearing_Necktie
090988.jpg
Given Label: No_Beard, Wearing_Hat, Wearing_Necktie, Eyeglasses
039745.jpg
Given Label: Eyeglasses, Wearing_Hat, Wearing_Necktie
Inconsistency in **Eyeglasses** Annotations
120309.jpg
Given Label: Eyeglasses, No_Beard, Smiling
053797.jpg
Given Label: No_Beard, Smiling
027391.jpg
Given Label: Eyeglasses, Wearing_Hat, No_Beard
138557.jpg
Given Label: No_Beard, Smiling

Figure 4: Examples of inconsistency in `Wearing_Hat`, `No_Beard`, or `Eyeglasses` annotations in the CelebA dataset. Note how for each pair of vertically-arranged similar images, one of these tags (e.g. `Wearing_Hat`) is annotated for the top image but not the bottom image, so one of their labels is clearly incorrect. Images shown are selected among the smallest (most likely erroneous) EMA label quality scores. These examples represent a small subset of the many issues in these tags that we observed throughout the CelebA dataset.

References

Carla E Brodley and Mark A Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999.

Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Jan-Christoph Klie, Bonnie Webber, and Iryna Gurevych. Annotation error detection: Analyzing the past and present for a more coherent future. *arXiv preprint arXiv:2206.02280*, 2022.

Johnson Kuan and Jonas Mueller. Model-agnostic label quality scoring to detect real-world label errors. In *ICML DataPerf Workshop*, 2022.

Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Nicolas M. Müller and Karla Markert. Identifying mislabeled instances in classification datasets. In *International Joint Conference on Neural Networks*, 2019.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, 2013.

Andrew Ng, Dillon Laird, and Lynn He. Data-centric ai competition, 2021. *URL https://https-deeplearning-ai. github. io/data-centric-comp*, 2021.

Curtis G. Northcutt, Tailin Wu, and Isaac L. Chuang. Learning with confident examples: Rank pruning for robust classification with noisy labels. 2017.

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Proceedings of the 35th Conference on Neural Information Processing Systems Track on Datasets and Benchmarks*, December 2021a.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021b.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Frederick Reiss, Hong Xu, Bryan Cutler, Karthik Muthuraman, and Zachary Eichenberger. Identifying incorrect labels in the CoNLL-2003 corpus. In *Proceedings of the 24th conference on computational natural language learning*, pp. 215–226, 2020.

Matthias Rottmann and Marco Reese. Automated detection of label errors in semantic segmentation datasets via deep learning and uncertainty quantification. *arXiv preprint arXiv:2207.06104*, 2022.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.

Wei-Chen Wang and Jonas Mueller. Detecting label errors in token classification data. *arXiv preprint arXiv:2210.03920*, 2022.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Appendix: Identifying Incorrect Annotations in Multi-Label Classification Data

A Extending Confident Learning to Flag Label Issues in Multi-labeled Data

This section details our proposed method to estimate the set of mislabeled examples in a multi-label dataset. For multi-class classification, Northcutt et al. (2021b) proposed *Confident Learning* (CL), an approach theoretically proven to identify the mislabeled examples under mild assumptions, even in the presence of asymmetric class-conditional label noise and an imperfectly trained classifier. Despite only requiring predicted probabilities from any trained classifier to identify label errors, Confident Learning has been shown to empirically outperform many more complex approaches (Northcutt et al., 2021b; Klie et al., 2022). Thus, we consider an extension of the CL framework to multi-label datasets.

A naive CL-extension might consider each observed combination of classes that co-occur for a particular example as a separate "class", which can result in up to 2^K "classes" and is statistically and computationally impractical.

Instead here we adopt the one-vs-rest perspective. We consider for each datapoint the binary task of whether each label applies or not. Specifically, for each of the K classes, we form a binary label b_i^k whether class k applies to example i or not, according to the given multi-label annotation. We extract the corresponding kth entry from the predicted class probabilities p_i (denoted as p_i^k) as an estimate of the probability that $b_i^k = 1$. For each class k, Confident Learning is independently applied to these binary labels and predicted binary probabilities p_i^k to flag examples with estimated wrong binary label b_i^k annotations. By combining the *union* of these binary errors detected over all classes $k = 1, ..., K$ for each example, we can report the subset of examples in the multi-label classification dataset estimated to contain an error in their annotation. This approach entails a straightforward extension of Confident Learning to multi-label settings. For each class k, this approach estimates the number of examples i incorrectly annotated as *belonging to* class k via the following equation:

$$\left| \left\{ \boldsymbol{x}_i \in \boldsymbol{X}_k \ : \ (1 - p_i^k) \geqslant \frac{1}{|\boldsymbol{X}_k^C|} \sum_{\boldsymbol{x}_j \in \boldsymbol{X}_k^C} (1 - p_j^k) \right\} \right| \qquad \text{(false positives)} \tag{3}$$

Here $\boldsymbol{X}_k$ denotes the subset of examples annotated as belonging to class k (i.e. where $b_i^k = 1$), and $\boldsymbol{X}_k^C$ denotes its complement (i.e. the examples j for which $b_j^k = 0$). Estimating the number of examples incorrectly annotated as *not belonging to* class k is done in a similar fashion, see the detailed description by Northcutt et al. (2021b).

Our multi-label extension of Confident Learning comes with the following theoretical guarantees. Our approach here closely models a binary special case of Confident Learning studied by Northcutt et al. (2017), and can be proven to exactly flag the examples with label errors as long as the classifier is *rank-separable* for each one-vs-rest binary problem (Northcutt et al., 2017). As long as the number of classes K is significantly lower than the number of examples in the dataset N (which is almost always the case to train an effective classifier), the theoretical guarantees for the estimation of mislabeled examples established by Northcutt et al. (2021b) also hold for the approach we consider here.

This binary approach to flagging multi-label errors does *not* require additional modeling of dependencies between classes because we assume the multi-label classifier producing p_i already reflects these dependencies in its predictions. When applying these methods, we caution *against* training the classifier in a one-vs-rest manner and instead recommend training a multi-label classifier on the multi-label annotations to obtain binary predicted probabilities p_i^k for each example (i.e., the class-probabilities for each example do not sum to 1). Even if the inter-class dependencies are imperfectly captured in p_i, this approach can accurately flag the examples whose labels are not entirely correct as long as the per-class binary probabilities p_i^k estimate the binary label b_i^k well enough to satisfy the conditions of Northcutt et al. (2021b). As with Confident Learning, the efficacy of our approach does depend on the accuracy of the trained classifier. In multi-label settings, a classifier that effectively models dependencies between classes will usually produce more accurate predictions.

B Additional Pooling Methods

Many alternative pooling methods could be applied to obtain a label quality score per example. Here we consider additional approaches evaluated in our benchmark study presented in Section 3.

Softmin pooling. Recall that pooling scores via a hard minimum implies s_i^* is solely determined by one of the s_i^k values for each example. An alternative way to take the other classes' scores into consideration while still emphasizing the minimum, as done by our EMA method, is to pool via a softer minimum-like operator. Recalling that $\mathbf{s}_i = (s_i^1, ..., s_i^K)$ denotes a vector of the per-class scores for example i, a simple way to pool using a softer minimum is via the softmax operator as follows:

$$s_i^* = \mathbf{s}_i \cdot \text{softmax}_\tau(1 - \mathbf{s}_i) = \frac{\sum_{k=1}^K s_i^k \cdot \exp\left(\frac{1-s_i^k}{\tau}\right)}{\sum_{k=1}^K \exp\left(\frac{1-s_i^k}{\tau}\right)} \tag{4}$$

Here $\tau > 0$ denotes the *temperature* of the softmax. We choose $\tau = 0.1$ which heavily emphasizes the smallest scores corresponding to the most suspicious class annotations. Larger values of τ did not perform as well in our benchmarks.

Log-transform Pooling. Alternatively, taking an arithmetic mean of the logarithm of all the scores should significantly emphasize the low-scoring class for a particular example, while still being accounting for the remaining classes' scores.

$$s_i^* = \frac{1}{K} \sum_{k=1}^{K} \log(s_i^k + \epsilon) \tag{5}$$

Here a positive infinitesimal, $\epsilon = 1e - 8$, is added to all scores to avoid scores of zero. This s_i^* corresponds to a common loss function used to train multi-label classifier models (log-likelihood). Much existing research has proposed using the training loss of each example as a label quality score (Klie et al., 2022; Müller & Markert, 2019), as done via this log-transform approach.

Cumulative Average of Bottom Scores. For subsequent pooling methods, we let $\hat{s}_i^1 \leqslant ... \leqslant \hat{s}_i^K$, denote the same values as $s_i^1, ..., s_i^K$ now sorted in increasing order. Another pooling method that emphasizes the smallest scores, but not only the minimum value, is to take an average of the J smallest scores for each example. Here we simply set $J = 2$ as larger values did not perform as well.

$$s_i^* = \frac{1}{J} \sum_{k=1}^{J} \hat{s}_i^k \tag{6}$$

Weighted Sum of Cumulative Averages. A more flexible variant of this method is to take cumulative averages across the per-class scores (again sorted in ascending order) for different values of J, and subsequently report an exponentially-weighted sum of these averages as the aggregated score.

$$s_i^* = \sum_{J=1}^{K} \sum_{k=1}^{J} \frac{\exp(1 - J)}{J} \hat{s}_i^k \tag{7}$$

Mean Simple Moving Average (SMA). Most of the presented pooling methods aim to emphasize lower per-class scores based on the intuition that these should matter more to detect examples i for which any of $b_i^1, ..., b_i^K$ are misannotated. De-emphasizing the other classes' scores helps mitigate nuisance variation in the scores for class annotations that are likely correct. Another way to mitigate nuisance variation may be to smooth the scores across different classes. We consider simple moving averages of the sorted scores with period $P < K$. Moving averages smooth out variation in the scores for adjacent classes (in the sort ordering), which are often both correctly annotated. These moving averages can then be mean-pooled as an overall label quality score.

$$s_i^* = \frac{1}{P(K-P+1)} \sum_{k'=P}^{K} \sum_{k=k'-P+1}^{k'} \hat{s}_i^k \tag{8}$$

Empirically, we found $P = 2$ outperformed larger values of P in our benchmarks.

C Details for Label Quality Score Benchmark

Metrics. We also consider a few other metrics to evaluate label quality scores, by comparing them against discrepancies between ground truth labels and noisy given labels:

- **AUPRC** is the area under Precision-Recall curve obtained by comparing estimated scores against a binary target. Note that **AUPRC** $=$ **AP** @ N where N is the number of examples in the dataset.
- **Average 3-Precision** @ T measures how well scores prioritize examples with at least 3 incorrect binary labels, among the bottom-T scoring examples.
- **Spearman Correlation** measures how well scores retrieve examples with many incorrect binary labels. It is the Spearman Correlation between the (complementary) label quality score and the number of tags b_i^k which are incorrectly annotated for example i.

Datasets. Next, we describe our *large/small* groups of multi-label classification datasets used for evaluation. The ten datasets in each group were each generated from the same underlying distribution with different random seeds. We produced each dataset via the `make_multilabel_classification` data generator from the `sklearn` package[2] (Pedregosa et al., 2011). This method produces bag of words features (our datasets had an expected word count of 500). Table 1 lists various differences between our *Large* and *Small* datasets.

We introduce class-wise label noise in the given labels by randomly generating class noise matrices with traces T_k for class k, described by eqs. (9) and (10):

$$Y_k = (1 - X_k)\left(1 - \frac{\exp\left(\frac{-(k_{\text{argsort}}-1)^2}{K}\right)}{2K}\right) \tag{9}$$

$$T_k = \max\{2Y_k, 2 - 2Y_k\} \tag{10}$$

where $X_k \sim \Gamma(\kappa, \theta)$ is drawn from a gamma distribution with $(\kappa, \theta) = (2, 0.01)$ and k_{argsort} is the index of X_k in $X = (X_1, \dots, X_K)$ after being sorted in ascending order. The expression in the latter parentheses in eq. (9) reweights the samples exponentially based on their relative ordering to emphasize classes with the worst label noise. Each example i is limited to at most 3 errors in the individual per class annotations $b_i^1, ..., b_i^K$.

Table 1: Summaries of the two groups of datasets used to benchmark label quality scores.

	Small	Large
Number of samples	5000	30000
Number of test samples	1000	7500
Number of features	3	20
Number of classes	4	50
Expected number of class assignments	2	5

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_multilabel_classification.html`

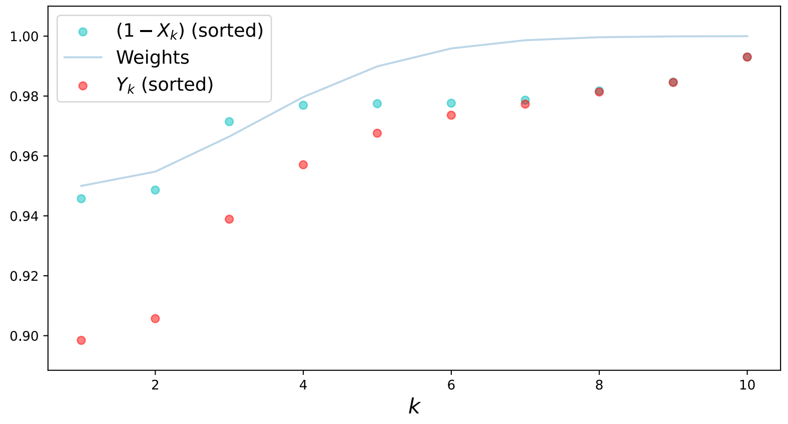


Figure 5: Average traces of noise matrices for a dataset with $K = 10$ classes, sampled via eq. (9).

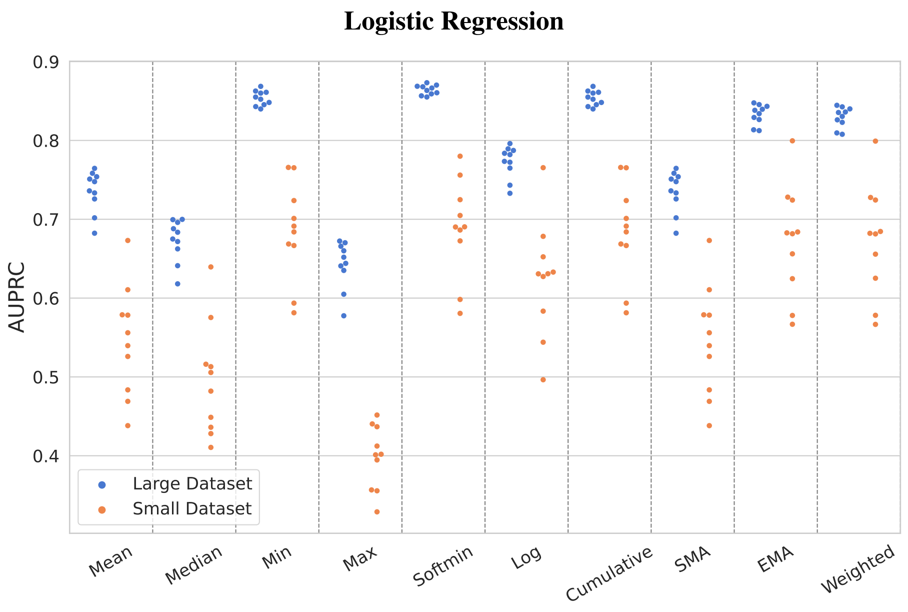
D Additional Results

Table 2: The average-per-class accuracies and Jaccard scores of different models trained and tested with either true or noisy labels for each group of datasets in our label quality scoring benchmark. Values reported are the mean over the 10 datasets in each group. In parenthesis is the corresponding standard deviation to the precision of the least significant digit of the mean value.

Datasets	**Classifier**	**Train labels**	**Test labels**	**Average accuracy**	**Jaccard score**
Small	Logistic regression	Noisy	Noisy	0.76(3)	0.62(4)
			True	0.81(4)	0.69(6)
		True	Noisy	0.75(3)	0.61(5)
			True	0.81(3)	0.69(6)
	Random forest	Noisy	Noisy	0.81(3)	0.69(5)
			True	0.88(4)	0.81(6)
		True	Noisy	0.82(3)	0.72(5)
			True	0.90(4)	0.86(6)
Large	Logistic regression	Noisy	Noisy	0.906(3)	0.30(2)
			True	0.922(2)	0.36(2)
		True	Noisy	0.907(3)	0.31(2)
			True	0.923(2)	0.38(2)
	Random forest	Noisy	Noisy	0.903(3)	0.25(2)
			True	0.919(2)	0.31(2)
		True	Noisy	0.903(3)	0.25(2)
			True	0.919(2)	0.31(2)

Tag	**Accuracy**
Eyeglasses	0.97
Wearing_Earrings	0.84
Wearing_Hat	0.97
Wearing_Necklace	0.87
Wearing_Necktie	0.93
No_Beard	0.92
Smiling	0.81

Table 3: Held-out accuracy for each tag (i.e. class) obtained by our CelebA multi-label classifier.



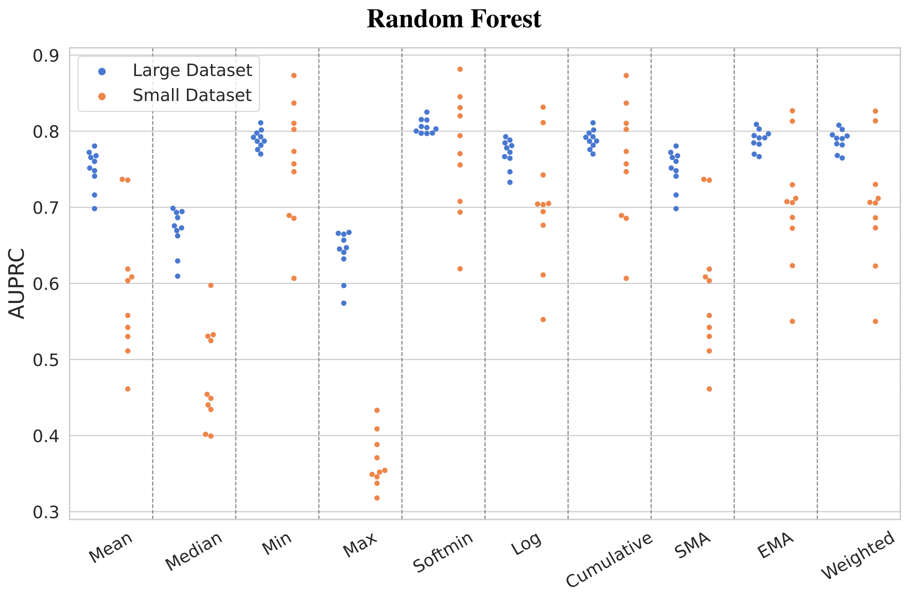
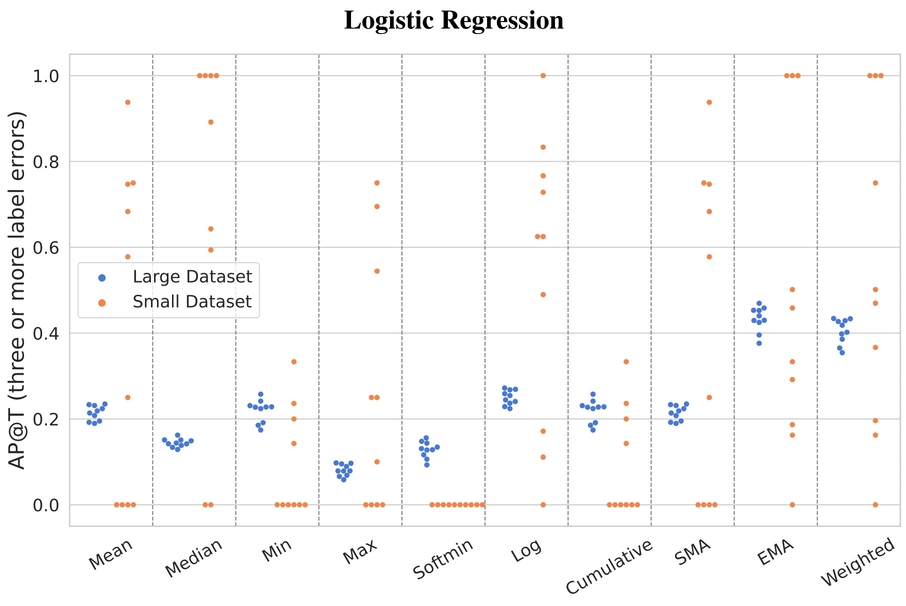


Figure 6: AUPRC for detecting mislabeled examples achieved by various overall label quality scores s_i^* for each dataset (shown as dot). 1We show results based on predicted class probabilities from both a Logistic Regression model and a Random Forest model.



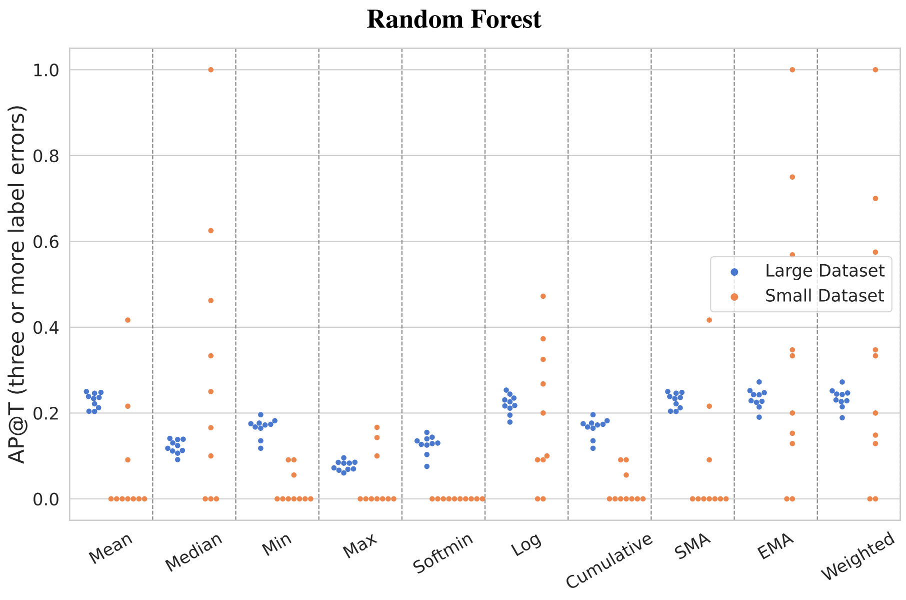
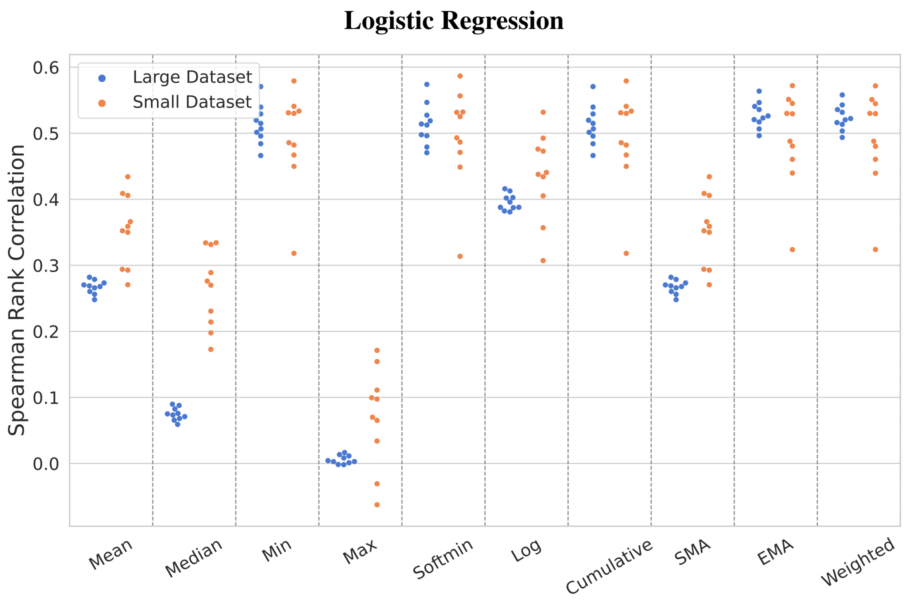


Figure 7: Average 3-Precision @ T achieved by various overall label quality scores s_i^*, where T is the number of mislabeled examples in each dataset. We show results based on predicted class probabilities from both a Logistic Regression model and a Random Forest model.



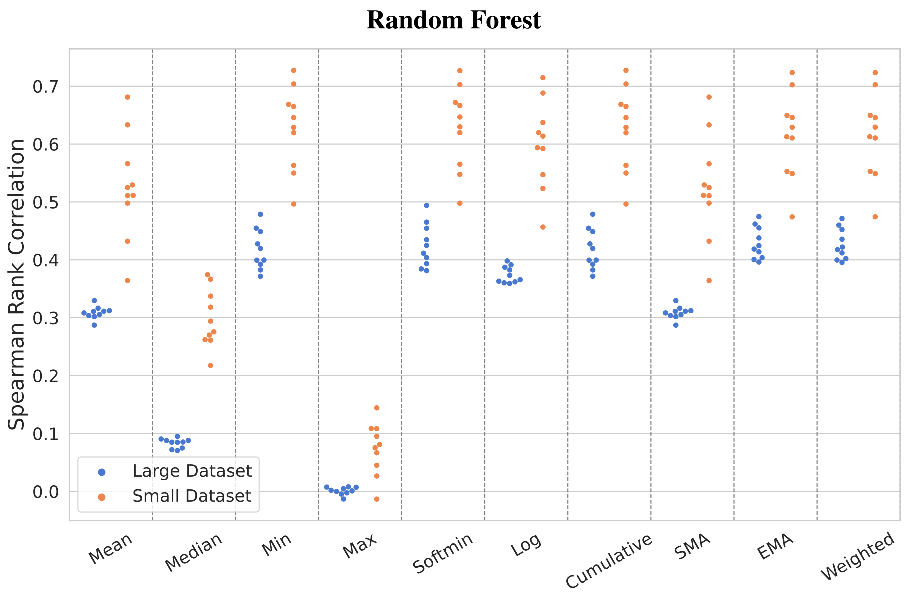
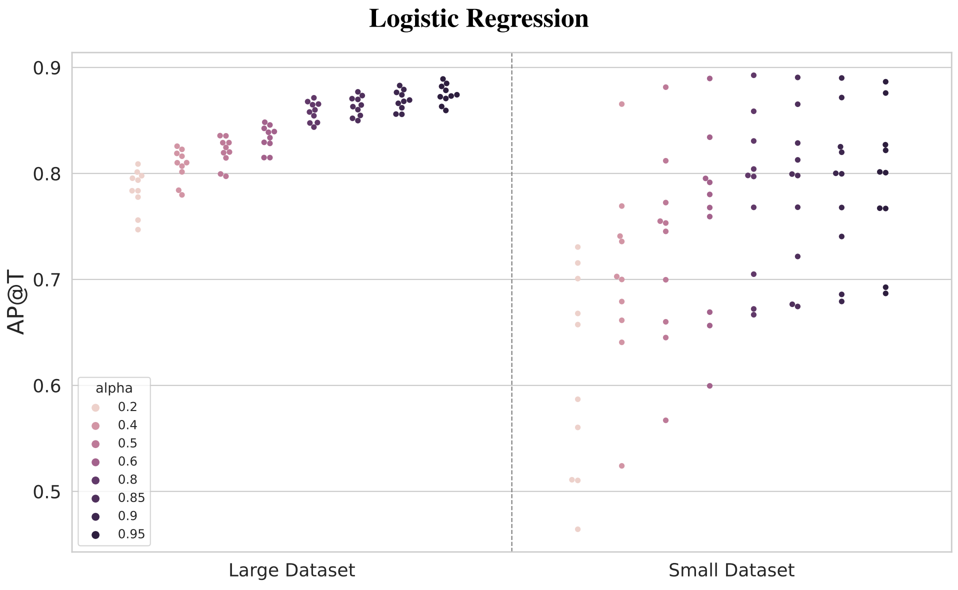


Figure 8: Spearman correlation between label quality scores s_i^* (produced via various methods) and number of class annotations $b_i^1, ..., b_i^K$ which are incorrect per example i. We show results based on predicted class probabilities from both a Logistic Regression model and a Random Forest model.



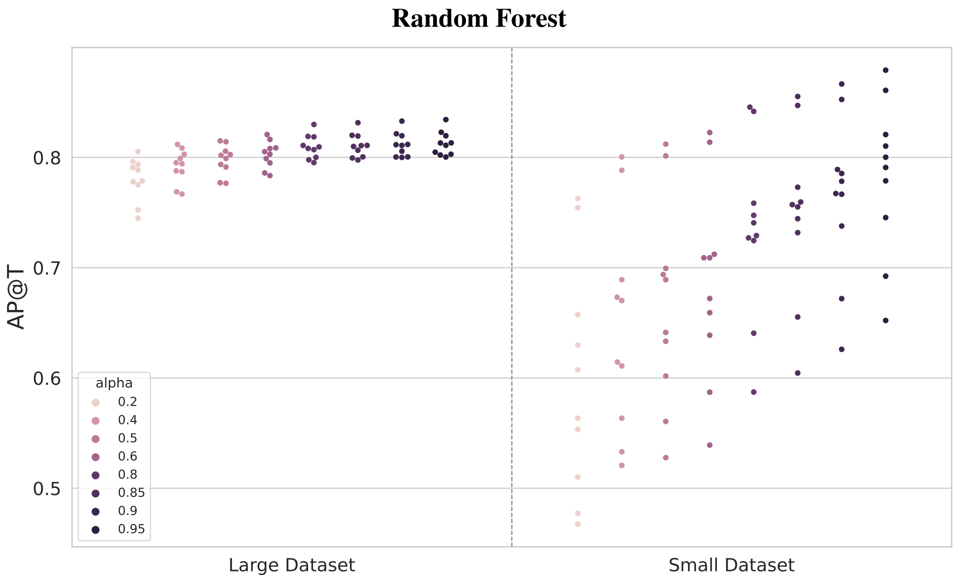
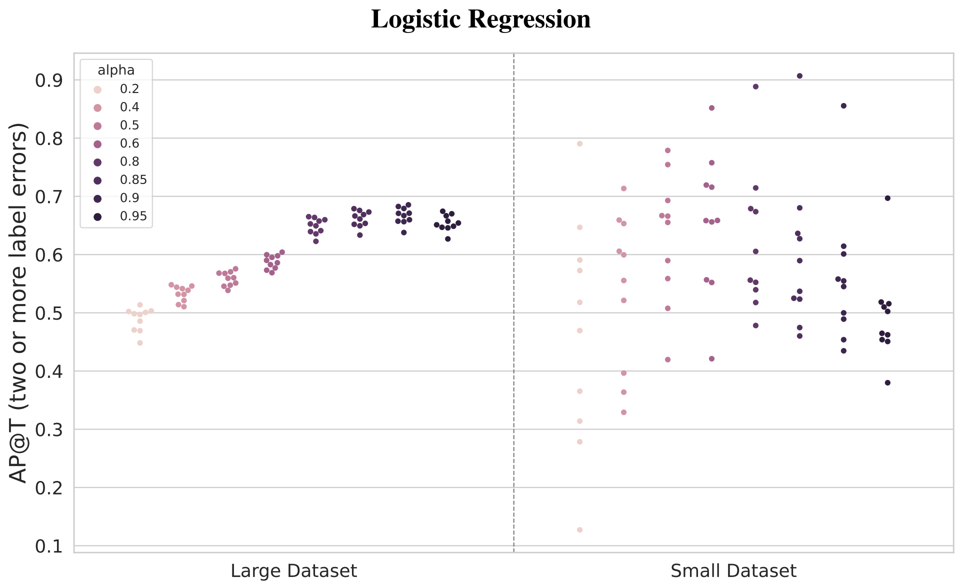


Figure 9: Average Precision @ T achieved by our EMA label quality scoring method with different values of α. Here T is the number of mislabeled examples in each dataset, and all examples with any sort of error in their label are counted as positive hits. We show results based on predicted class probabilities from both a Logistic Regression model and a Random Forest model.



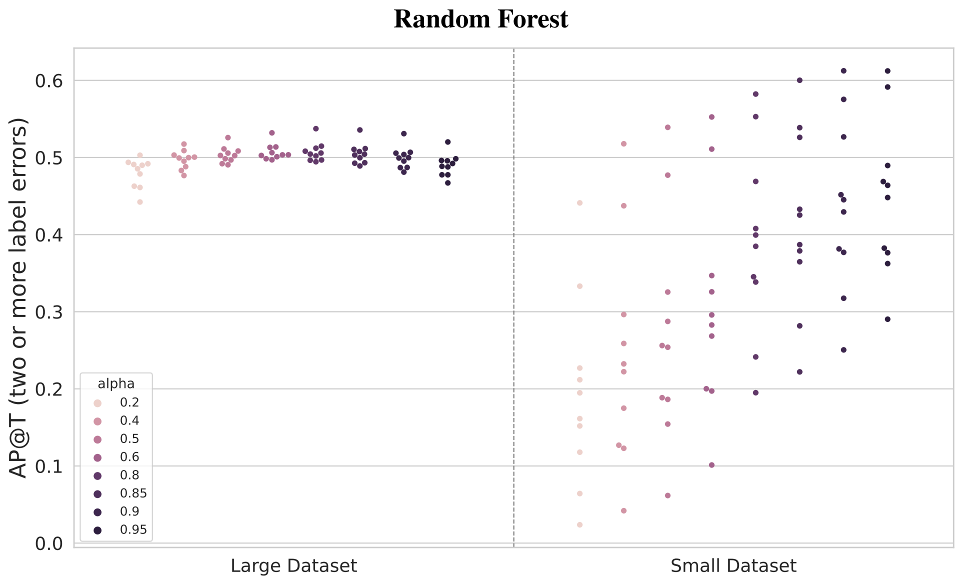
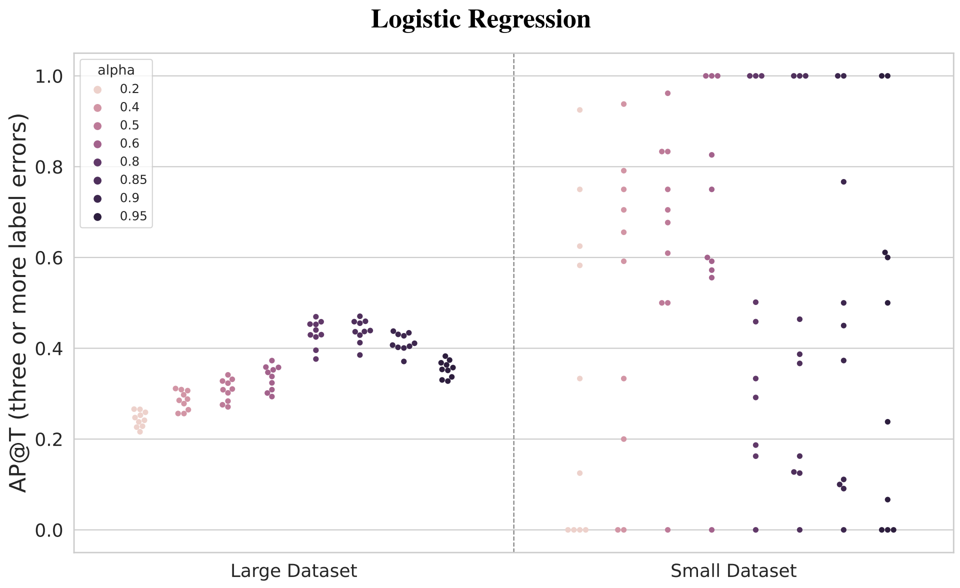


Figure 10: Average 2-Precision @ T achieved by our EMA label quality scoring method with different values of α. In 2-Precision, an example is only counted as a positive hit if its label contains at least 2 misannotated classes. T is the number of mislabeled examples in each dataset. We show results based on predicted class probabilities from both a Logistic Regression model and a Random Forest model.



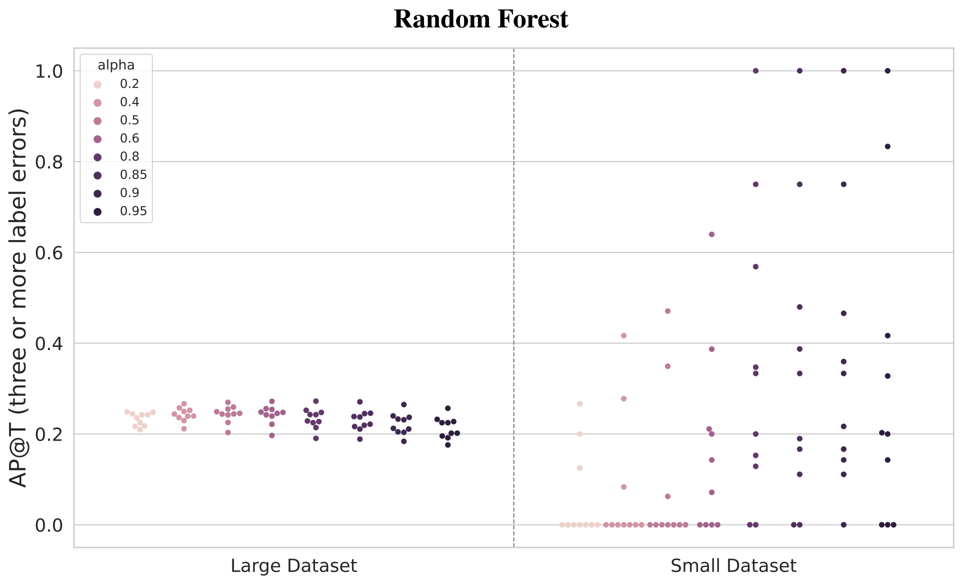
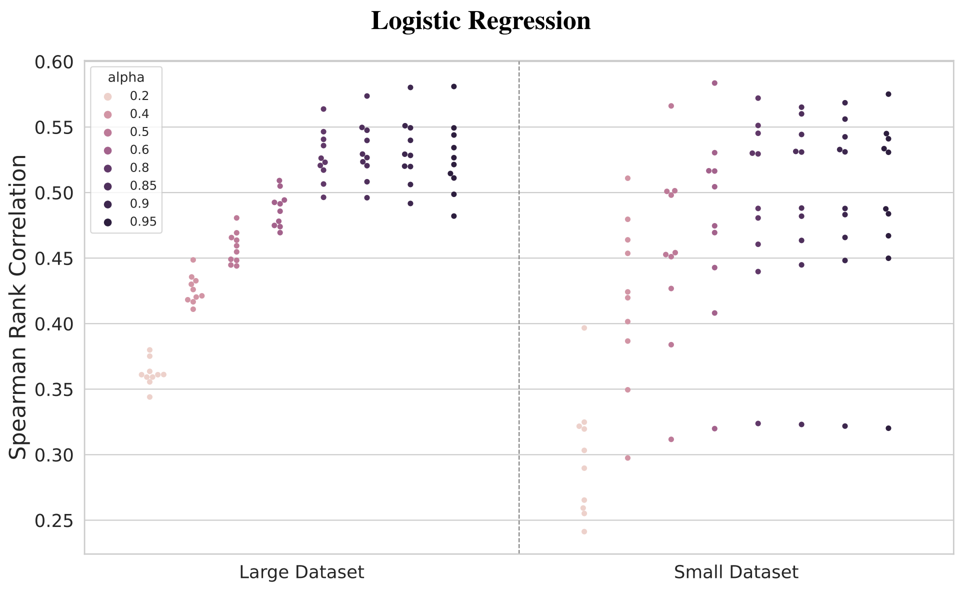


Figure 11: Average 3-Precision @ T achieved by our EMA label quality scoring method with different values of α. In 3-Precision, an example is only counted as a positive hit if its label contains at least 3 misannotated classes. T is the number of mislabeled examples in each dataset. We show results based on predicted class probabilities from both a Logistic Regression model and a Random Forest model.



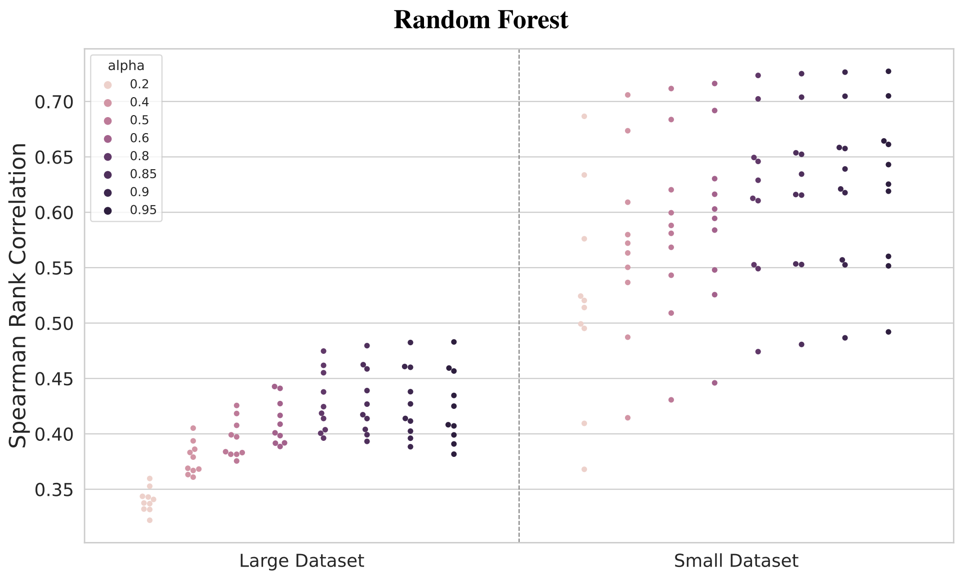


Figure 12: Spearman correlation between label quality scores s_i^* and number of class annotations $b_i^1, ..., b_i^K$ which are incorrect per example i. The scores s_i^* are produced via our EMA method with different values of α. We show results based on predicted class probabilities from both a Logistic Regression model and a Random Forest model.