# Engineering Uncertainty Representations to Monitor Distribution Shifts

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

In some classification tasks, the true label is not known until months or even years after the classifier prediction time. Once the model has been deployed, harmful dataset shift regimes can surface. Without cautious model monitoring, the damage could prove to be irreversible when true labels unfold. In this paper, we propose a method for practitioners to monitor distribution shifts on unlabeled data. We leverage two representations for quantifying and visualizing model uncertainty. The *Adversarial Neighborhood Analysis* assesses model uncertainty by aggregating predictions in the neighborhood of a data point and comparing them to the prediction at the single point. The *Non-Conformity Analysis* exploits the results of conformal prediction and leverages a decision tree to display uncertain zones. We empirically test our approach over scenarios of synthetically generated shifts to prove its efficacy.

## 1 Introduction

In classification problems, the true label is sometimes not known until months or even years after a Machine Learning (ML) classifier prediction. For instance, with a 12-month default horizon, it is not possible to assess the true performance of a credit granting model before one year. In selecting a cancer treatment for a patient, a model can predict therapy response or resistance [1]. Once again, the treatment outcome will be confirmed with a certain lag. After model deployment, dataset shifts may arise when training (the Source set $S$) and real-world data (the Target set $T$) joint distributions are different. Formally, for a set of covariates $\boldsymbol{x}$ and label $y$, dataset shift arises when $p_S(\boldsymbol{x}, y) \neq p_T(\boldsymbol{x}, y)$. In the context of unlabeled Target data, an absence of monitoring could wreak havoc if the classifier made predictions for several months under harmful covariate shifts.

**Scope and contribution** In this paper, we propose a method for practitioners to monitor distribution shifts. We assume that the ML classifier produces class probability estimates. This method aims to meet the following goals: (i) quantify the change in model uncertainty between the Source and Target data, (ii) visualize the source of uncertainty. To this end, our method creates and exploits two representations from unlabeled data. The *Adversarial Neighborhood Analysis (ANA)* assesses model uncertainty by aggregating predictions in the neighborhood of a data point and comparing them to the prediction at the single data point. *Disagreement* zones are labeled in a specific way. The change in uncertainty between $S$ and $T$ can be measured and a decision tree is used in order to visualize uncertain areas. The *Non-Conformity Analysis (NCA)* exploits the results of conformal prediction and leverages a tree to display uncertain zones.

Covariate shifts are defined as $p_S(y|\boldsymbol{x}) = p_T(y|\boldsymbol{x})$ and $p_S(\boldsymbol{x}) \neq p_T(\boldsymbol{x})$ [10]. Prior probability shift is defined as $p_S(\boldsymbol{x}|y) = p_T(\boldsymbol{x}|y)$ and $p_S(y) \neq p_T(y)$. These changes can exacerbate model uncertainty and deteriorate its performance [12, 14]. Uncertainty quantification is central for detecting dataset

Submitted to Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

shifts. Epistemic uncertainty is connected to the model uncertainty due to insufficient training data; it can be harnessed to detect out-of-distribution samples [15]. Uncertainty quantification is not intrinsic to discriminative classifiers $\hat{p}(y|\boldsymbol{x})$, even though they produce class probability estimates. Lastly, conformal prediction is an uncertainty quantification method which calibrates conformal scores on a labeled dataset to produce prediction sets on unlabeled data with a given coverage [2, 5, 13].

## 2 Method for quantifying and visualizing uncertainty

**Notations** We consider a Source dataset $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ with features or covariates $\boldsymbol{x}_i \in \mathbb{R}^d$ and discrete label $y_i \in \Upsilon = \{1, 2, \ldots, C\}$ for $C$-class classification tasks. The samples are drawn i.i.d. from true unknown distribution $p_S(\boldsymbol{x}, y)$. The ML classifier $\hat{p}_y(\boldsymbol{x}) = \hat{p}(y|\boldsymbol{x})$ estimates the true unknown class probabilities $p_y(\boldsymbol{x}) = \mathbb{P}(Y = y|\boldsymbol{X} = \boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{R}^d$, for any $y \in \Upsilon$. A predicted label $\hat{y}$ is based on the argmax of the vector of predicted class probabilities. At the time of monitoring, we assume that the Target set $T = \{\boldsymbol{x}_i\}_{i=n+1}^{p}$ is unlabeled. Lastly, the Source data is randomly partitioned into two shares, for training and calibration: $(S_{Train}, S_{Calib})$.

**Technique 1: Adversarial Neighborhood Analysis (ANA)** The Adversarial Neighborhood Analysis aims to identify the increase in model uncertainty and characterize it. It observes the stability of model predictions when inferring in a small region around a Target data point. We sample data points in a small hypercube centered at a given Target example and examine whether the hypercube majority vote differs from the prediction at the single data point (i.e., hypercube center). In fact, we want to identify the data points where the ML classifier predictions are most unstable. Assuming the ML classifier has been trained on $S_{Train}$, we apply this technique on $S_{Calib}$ and on the Target data $T$. We define the hypercube of center $c$ and length $R$ as $H(c, R) = \{\boldsymbol{x}|\text{feature } x^{(j)} \in [0, 1] \text{ and } |x^{(j)} - c^{(j)}| \leq R, \forall j = 1, ..., d\}$. For each example of $S_{Calib}$ and then of the Target data, we compute the proportion of instances where the hypercube majority vote dissents from the predicted label $\hat{y}$ at the example alone. We call this metric the disagreement rate ($DR$). It indicates the percentage of uncertain outputs:

$$DR(T) = \frac{|\{\boldsymbol{x}|\boldsymbol{x} \in T \text{ and mode}(\{argmax_y \hat{p}_y(\boldsymbol{z}), \boldsymbol{z} \in H(\boldsymbol{x}, R)\}) \neq \hat{y}\}|}{|T|}$$

If the disagreement rate on the Target data is significantly greater than on $S_{Calib}$, we can suspect that the ML classifier predictions are less robust in the Target domain. For instance, if $DR$ doubles between the domains, the model is twice as much uncertain. Most importantly, we can easily identify the Target samples where disagreement occurs, and thus where model uncertainty lies. In fact, using the disagreement as a binary variable, 1 being the label for dissent, we can construct a decision tree classifier, called *characterization tree*, to pinpoint the most uncertain regions.

**Technique 2: Non-Conformity Analysis (NCA)** We propose the Non-Conformity Analysis, an uncertainty representation tool, which exploits the output of conformal prediction. We aim to visualize in what regions of the Target domain the ML classifier is most uncertain. NCA reveals the paths leading to uncertainty in the Target domain. To this end, a decision tree is trained directly on the Target dataset using values produced by conformal prediction as labels. For each Target sample, conformal prediction generates a prediction set $\hat{C}_\alpha$ estimated with a given error tolerance $\alpha \in [0, 1]$. $\hat{C}_\alpha$ is then at least $1 - \alpha$ likely to include the true label. $1 - \alpha$ is called the coverage. To generate the prediction sets, we proceed as follows: (i) *ML classifier training*: we partition the Source data into $S_{Train}$ and $S_{Calib}$. The ML classifier is trained on $S_{Train}$. $S_{Calib}$ contains $m$ instances. (ii) *Conformal calibration*: we compute the conformal score $s_i$ as one minus the predicted probability of the true label on each instance $\boldsymbol{x}_i$ of $S_{Calib}$, that is $s_i = 1 - \hat{p}_{y_i}(\boldsymbol{x}_i)$ where $y_i$ is the true label. Equipped with our conformal scores, we then compute $\hat{q}$, defined as the $(1 - \alpha) \times \frac{m+1}{m}$ corrected quantile of $s_1, \ldots, s_m$. (iii) *Conformal prediction*: for each new Target sample $\boldsymbol{x}$, we produce a prediction set $\hat{C}_\alpha(\boldsymbol{x}) = \{y|1 - \hat{p}_y(\boldsymbol{x}) \leq \hat{q}\}$. The latter can include one label, several labels (ambiguity), or can be empty (no label assigned). For a given $\alpha$, ambiguity occurs when the model hesitates and shares the predicted probabilities between several classes, i.e., several $\hat{p}_y(\boldsymbol{x})$ may reach the required level to be included into $\hat{C}_\alpha$. The set is empty when the model hesitates but with no predicted class probabilities fulfilling the condition. These sets are used as labels to train a tree on the Target domain. Sets with

Table 1: ANA $DR(T)/DR(S_{Calib})$ results by use case and shift scenario, averaged over 10 dataset split seeds. Numbers between parentheses are standard deviations.

| Use case / Scenario | Adverse | Benign | Standard |
|---|---|---|---|
| 1.Default of credit card clients | 5.53 (1.78) | 0.73 (0.26) | 1.04 (0.21) |
| 2.Lending Club, default | 7.03 (5.71) | 1.71 (0.55) | 1.12 (0.57) |

only one predicted label are labeled as 0 (e.g., $\{1\}, \{2\}$), while the empty set or sets with several predicted labels are uncertain and annotated as 1 (e.g., $\{\}, \{1, 2, 3\}$).

# 3 Experiments and discussion

## 3.1 Settings

The Target dataset including the true labels is generated according to 3 different scenarios: (i) Standard scenario: $p_T(\boldsymbol{x}, y) = p_S(\boldsymbol{x}, y)$, i.e., no distribution shift. (ii) Benign scenario: covariate shift occurs; the label proportions remain pretty similar between the Source and Target datasets, and the ML classifier accuracy remains stable as well. (iii) Adverse scenario: covariate shift occurs; the label proportions differ between the Source and Target data, and the ML classifier accuracy on the Target data is adversely affected. Each experiment is run over these 3 scenarios and 10 seeds.

**Data and ML classifiers** 3 use cases are proposed. The first 2 use cases are binary classification tasks, with the prediction of credit default. The third use case is a multiclass classification task with text data (20 topics). First, the default of credit card clients case is a tabular dataset with 30k instances and 22 variables [17, 16]. Second, the data extract from Lending Club corresponds to 2018 accepted loans [4], with 56k observations and 23 variables. Lastly, we randomly extract 10k posts from the 20 Newsgroups dataset [8]. For use case 1, the ML classifier is based on CatBoost [11]. For use case 2, the ML classifier is a neural network with 2 fully connected hidden layers with drop-out and is trained using Adam optimizer [7]. Use case 3 employs a bidirectional Long Short-Term Memory (LSTM) model [6]. It is built with input sequences of length $150$, an embedding layer with $20000$ words and dimension of $64$, a bidirectional LSTM, a fully connected layer with drop-out and a final layer using softmax activation (20 classes). It is trained using Adam optimizer.

**Shift generation technique** Because creating synthetic data through input perturbations can produce label inconsistencies, we choose to not modify the original datasets. Dataset shifts are generated with k-means clustering approach [9] using $S \cup T$ (covariate) data. Instances within each cluster will have their proper covariate distribution and some of the clusters will have distinct label proportions. The baseline scenario results from sampling (without replacement) with a constant proportion over all the clusters. The benign scenario is constructed by sampling (without replacement) heterogeneously from the clusters for the Source and Target data. However, the label proportions are controlled so that they remain pretty close between the Source and Target domains. In the adverse scenario, the label proportions greatly differ between the two datasets by sampling differently from distant clusters.

## 3.2 Results

For ANA, the experiments are run with 500 instances randomly drawn from $S_{Calib}$ and with a Target volume of 500. We normalize continuous features using min-max scaling. For each data point, we uniformly sample $1000$ elements at random from $H(c, R = 0.1)$. Producing small perturbations around the hypercube center is straightforward for numerical variables. For the set of categorical features, we use one-hot encoding (OHE). To generate meaningful "in-distribution" perturbations for OHE variables, we first group $S_{Calib}$ instances into $1/R$ clusters using k-means and follow a similar approach for $T$. OHE values are then drawn from the same cluster as the hypercube center. Table 1 demonstrates that the disagreement rate ratios $DR(T)/DR(S_{Calib})$ greatly increase in the adverse case: the ML classifier decisions are more uncertain. This technique is relevant to detect potentially harmful shifts. Using the disagreement as a binary variable, 1 being the label for disagreement, we can construct a decision tree. This tool is appropriate to pinpoint the most uncertain regions as the leaf in the left sub-tree of Figure 1, which displays 92.9% disagreement. Human-in-the-loop [3] with

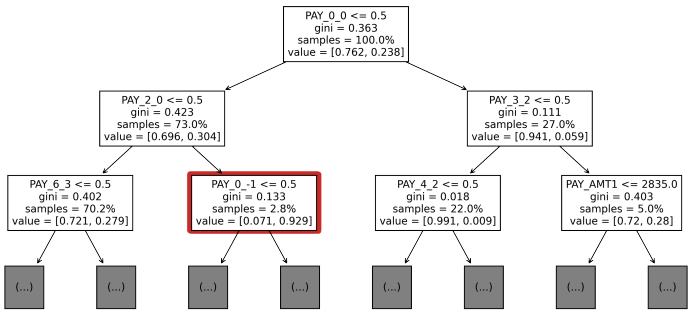


Figure 1: Characterization tree for use case 1, based on Adversarial Neighborhood Analysis.

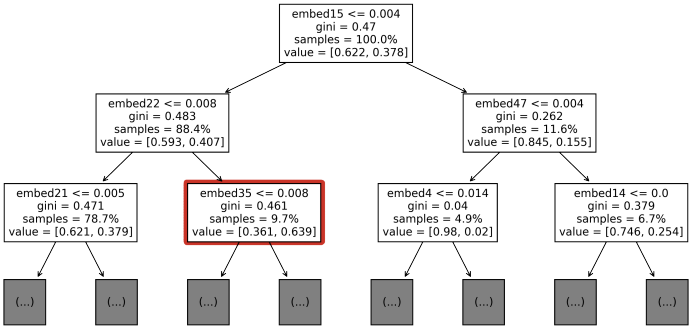


Figure 2: Non-Conformity Analysis with word embeddings, based on use case 3.

arbitration in the decision process helps to mitigate the risk of incorrect automated predictions for new samples that follow the identified patterns, i.e., new examples which would fall in the risky leaves. Another option would be to reduce the ML classifier application perimeter to feature regions where the model is less uncertain. It is worth noting that the results were not conclusive in the multiclass case and are not displayed here. This remains to be seen.

For NCA, we train a decision tree on the outputs of the conformal prediction in the Target domain, split into certain and uncertain sets. We test various Target volumes between 500 and 5000. Figure 2 displays the tree classifying 20 Newsgroups posts by uncertainty leveraging the word embeddings learned by the LSTM model (Target volume of 1000). The model is pretty uncertain globally, but one leaf includes 63.9% uncertain model outcomes according to conformal prediction with 60% coverage. We select the most frequent tokens that are present in the posts that fall into that leaf (token list $U$). We follow a similar approach for the rest of the posts (token list $C$). Among this selection, we then display the tokens that are exclusively in list $C$ and those exclusively in list $U$. Among the most frequent tokens of list $U$, we recognize words related to political or religious topics, such as $bill$ or $believe$. This is in line with the applied adverse scenario where those topics turn out to be mostly present in the Target domain. If the ML classifier outputs are used by a downstream model, we would recommend expert users to confirm the predicted labels for the posts that fall in that risky leaf.

## 4 Conclusion

We presented two practical monitoring techniques to assess and characterize model uncertainty in the context of distribution shifts. We have shown the relevance of these techniques on various types of ML classifiers and use cases. These tools can be employed for tasks with human interaction, in order to verify or confirm the predicted labels in risky zones. Displaying model's uncertainty and weak spots also serves the transparency goal. This will reinforce the trust and responsibility of model designers and users. Future work will focus on testing these tools on additional use cases (e.g., image classification) and on investigating situations where some of these techniques would fail.

## References


[1] George Adam, Ladislav Rampášek, Zhaleh Safikhani, Petr Smirnov, Benjamin Haibe-Kains, and Anna Goldenberg. Machine learning approaches to drug response prediction: challenges and recent progress. *NPJ precision oncology*, 4(1):1–10, 2020.

[2] Anastasios Nikolas Angelopoulos, Stephen Bates, Michael I. Jordan, and Jitendra Malik. Uncertainty sets for image classifiers using conformal prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[3] Lorrie Faith Cranor. A framework for reasoning about the human in the loop. In *Usability, Psychology, and Security, UPSEC'08, San Francisco, CA, USA, April 14, 2008, Proceedings*. USENIX Association, 2008.

[4] Nathan George. Lending club loan data. `https://www.kaggle.com/wordsforthewise/lending-club`, 2018. Accessed: 2022-06-01.

[5] Chirag Gupta, Arun K Kuchibhotla, and Aaditya Ramdas. Nested conformal prediction and quantile out-of-bag ensemble methods. *Pattern Recognition*, 127:108496, 2022.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[8] Ken Lang. The 20 newsgroups text dataset. `https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html`, 1995. Accessed: 2022-06-01.

[9] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982.

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

[11] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[12] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[13] Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems*, volume 33, pages 3581–3591, 2020.

[14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[15] Junjiao Tian, Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Exploring covariate and concept shift for out-of-distribution detection. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[16] UCI. Default of credit card clients data set. `https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients`, 2016. Accessed: 2022-06-01.

[17] I-Cheng Yeh and Che-Hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36 (2):2473–2480, 2009.