Institutions

1 UNIVERSITÀ DI TORINO https://www.unito.it/

2 RuleX https://www.rulex.ai/

The 38th Annual AAAI Conference on Artificial Intelligence

Using Stratified Sampling to Improve LIME Image Explanations

Muhammad Rashid[1], Elvio G. Amparore[1], Enrico Ferrari[2], Damiano Verda[2]

Fundings

EU Horizon-2020 ECSEL-JU project

NEXT PERCEPTION

https://nextperception.eu/

Abstract

PROBLEM

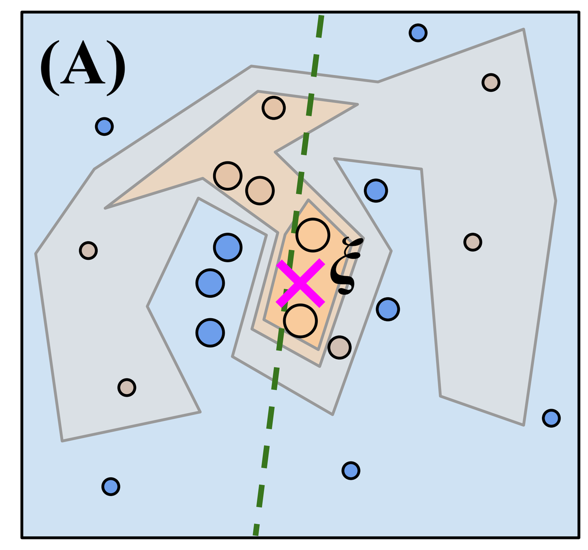
- LIME Image employs Monte Carlo sampling to generate synthetic neighborhoods
- The synthetic neighborhood aims to resemble a point cloud around the explained instance ξ, with perturbed samples varying in distance from ξ: some closer, other further away.
- However, the use of the Bernoulli distribution B with coefficient ½ concentrates the probability mass at the distribution center, allocating close-to-zero probability to the tails.
- Consequently, the synthetic neighborhood tends to look like a hypersphere positioned halfway between the original input and the fully-masked image.
- This can lead to a significant under-representation of the neighborhood, as very few synthetic samples closely resemble the original image.

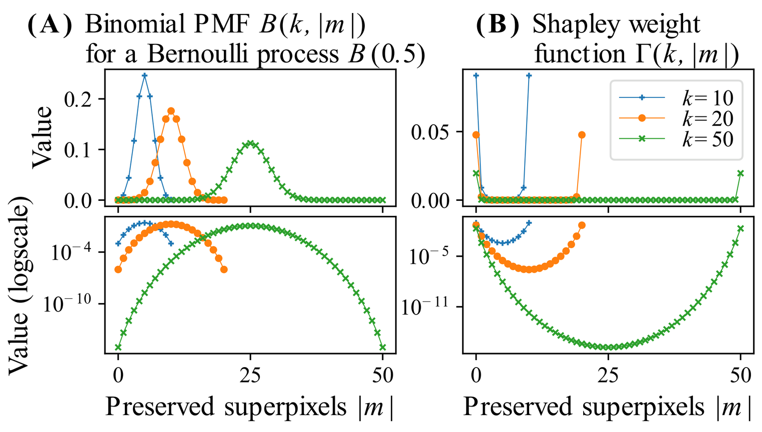
PROPOSAL

- Integrate a stratified sampling approach into LIME equations to overcome the undersampling issues.
- Samples are drawn from the entire sampling space of the possible perturbed inputs, divided by strata.
- Add weight factors to samples, based on the frequency of the strata they belong to, to keep the estimator unbiased.

The problem with unbiased Bernoulli distribution

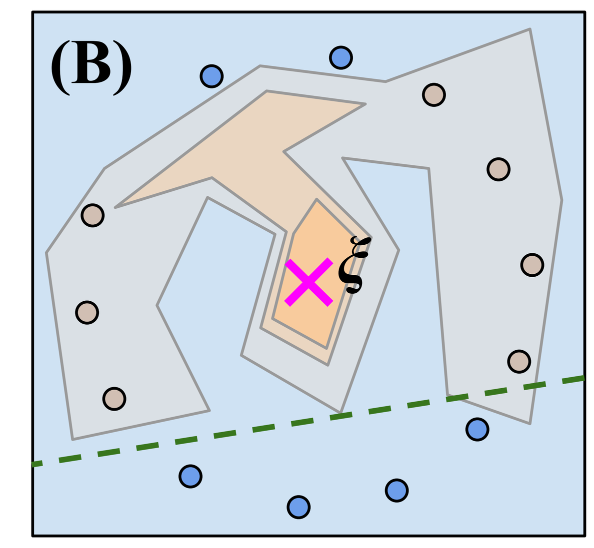
Ideally, the synthetic neighborhood $N(\xi)$ should provide a "good enough" coverage of the variations around ξ.





Samples at the tails of the Bernoulli distribution $B(0.5)$ are more rare than samples at the center of the distribution.

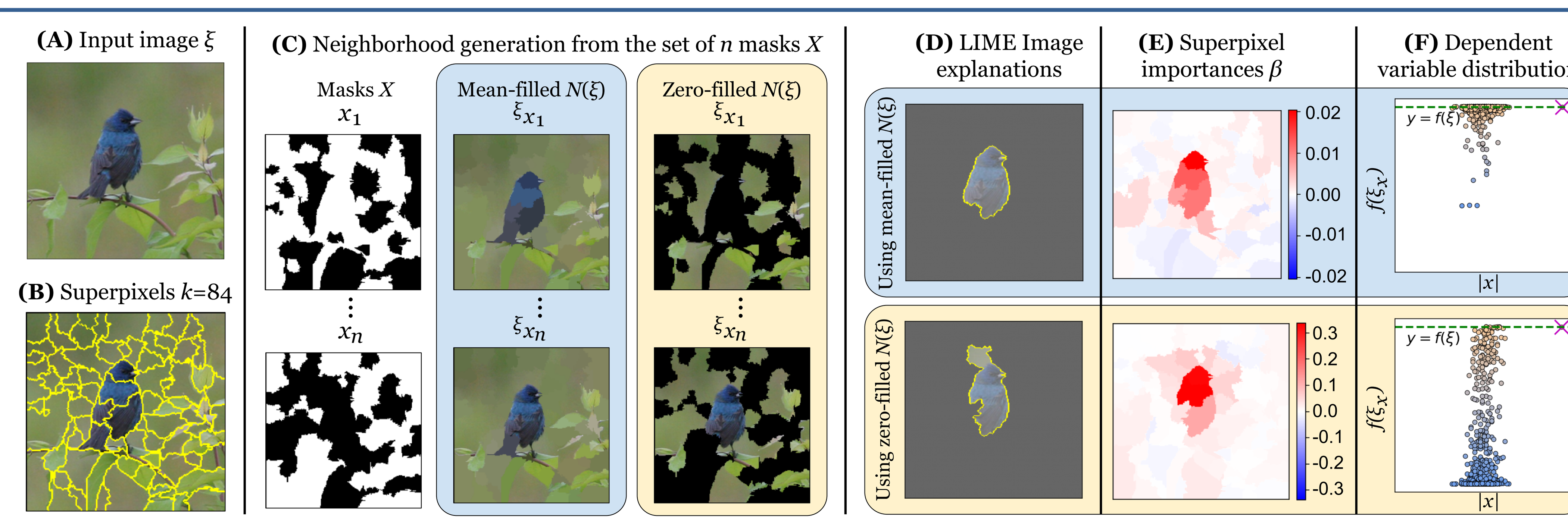
In practice the synthetic neighborhood $N(\xi)$ sampled by LIME Image looks like an hypersphere, with ver few to no samples close to the input image ξ.



How standard LIME Image works

NOTATION

- Initial image $\xi \rightarrow$ divided into k superpixels
- Superpixel masking: $x \in \{0,1\}^k$ generates a perturbed image ξ_x
- In LIME Image, masks are sampled using an unbiased Monte Carlo strategy: $x[i] \sim B(0.5), \quad 1 \leq i \leq k$ where $B(p)$ is a Bernoulli-distributed random variable having probability $p = 0.5$
- $N(\xi)$ = synthetic neighborhood of n perturbed images
- Dependent variables: $Y = \{ f(\xi_x) \mid \xi_x \in N(\xi) \}$



Dependent variable undersampling

Evaluation metrics

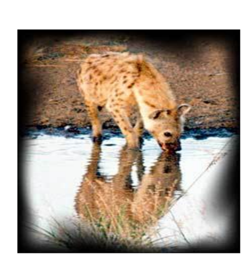
- Coefficient of Variation of the explanation β
  $CV(\beta) = \frac{\sigma_\beta}{\mu_\beta}$
- Range Coverage of the values of the dependent variable Y in the synthetic neighborhood.
  $RC(Y) = \frac{IQR_{1-99}(Y)}{f(\xi)}$

Observations

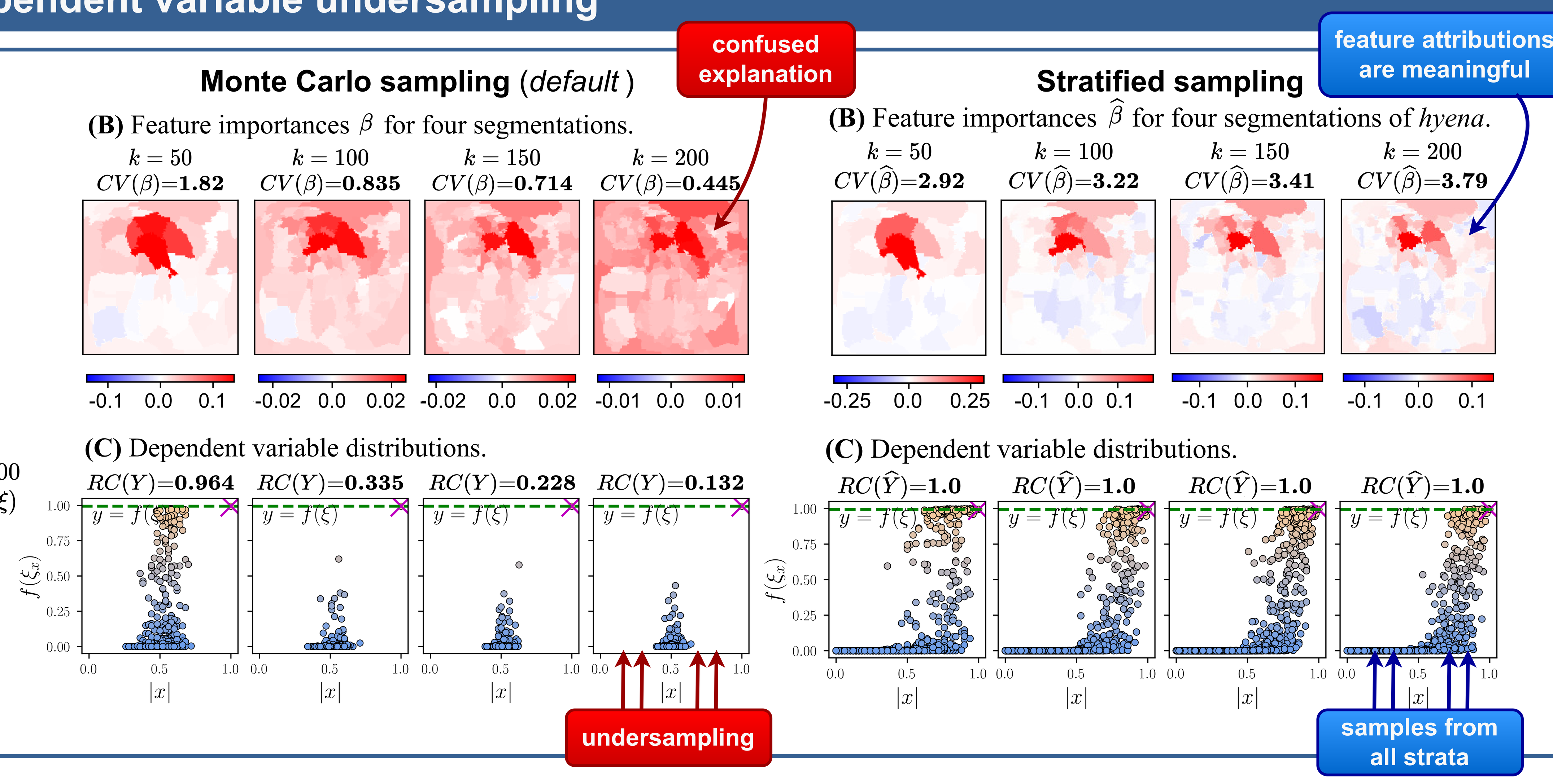
- Almost no sample is close to ξ.
- Samples drawn from $B(p)$ have ~50% of the superpixels masked.
- Under-representation of the local behaviour of the black-box model f.

Stratified sampling

- Consider a partitioning stratified on the number of masked superpixels.
- $\mathcal{X}^{(i)}$ = set of possible masks having $|x| = i$
- Stratum i size is known a priori:
  $|\mathcal{X}^{(i)}| = \binom{k}{i}, \qquad 0 \leq i \leq k$
- Oversampled probability
  $Prob\{x \in \mathcal{X}^{(i)} \mid x \in \widehat{X}\} = \frac{1}{k+1}$
- Adjustment factors
  $adj(i) = \frac{Prob\{x \in \mathcal{X}^{(i)} \mid x \in X\}}{Prob\{x \in \mathcal{X}^{(i)} \mid x \in \widehat{X}\}} = \frac{(k+1)\binom{k}{i}}{2^k}$



Input image ξ
Class: *hyena*
Probability: 99.46%
num_samples $n = 1000$
Using mean-filled $N(\xi)$



Monte Carlo vs Stratified Sampling

Monte Carlo sampling

- LIME Image uses a simple linear homoscedastic model
  $Y = X \cdot \beta + \epsilon$
- The explanation coefficients β results from
  $\beta = (X^\intercal W X)^{-1} X^\intercal W Y$

Stratified sampling

- β coefficients may vary by stratum.
- We adopt instead a mixture model
  $\widehat{Y}^{(i)} = \widehat{X}^{(i)} \cdot \widehat{\beta}^{(i)} + \hat{\epsilon}^{(i)}$
- The explanation coefficients β results from
  $\widehat{\beta} = (\widehat{X}^\intercal \widehat{W} \widehat{X})^{-1} \widehat{X}^\intercal \widehat{W} \widehat{Y}$
- $\widehat{W}$ accounts for the adjustment factors that correct the bias introduced by oversampling the distribution tails.

Impact of stratified sampling in LIME Image

Case (A): The mean and variance of $\widehat{\beta}^{(i)}$ are independent from the strata.

- weighted regression model is not needed.
- Monte Carlo and stratified sampling should behave similarly.
- Unlikely to happen using complex black-box machine learning models.

Case (B): The mean and variance of $\widehat{\beta}^{(i)}$ varies by stratum.

- weighted regression model is highly advisable (DuMouchel and Duncan 1983).
- Monte Carlo will perform badly, stratified sampling is relevant.
- Common scenario for complex models and/or large num. of superpixels.

Evaluation & conclusions

- About 1 image out of 5 in the ImageNet Object Localization Dataset suffers from severe undersampling using the default Monte Carlo sampling of LIME-Image
- Misbehaviours are corrected using stratified sampling

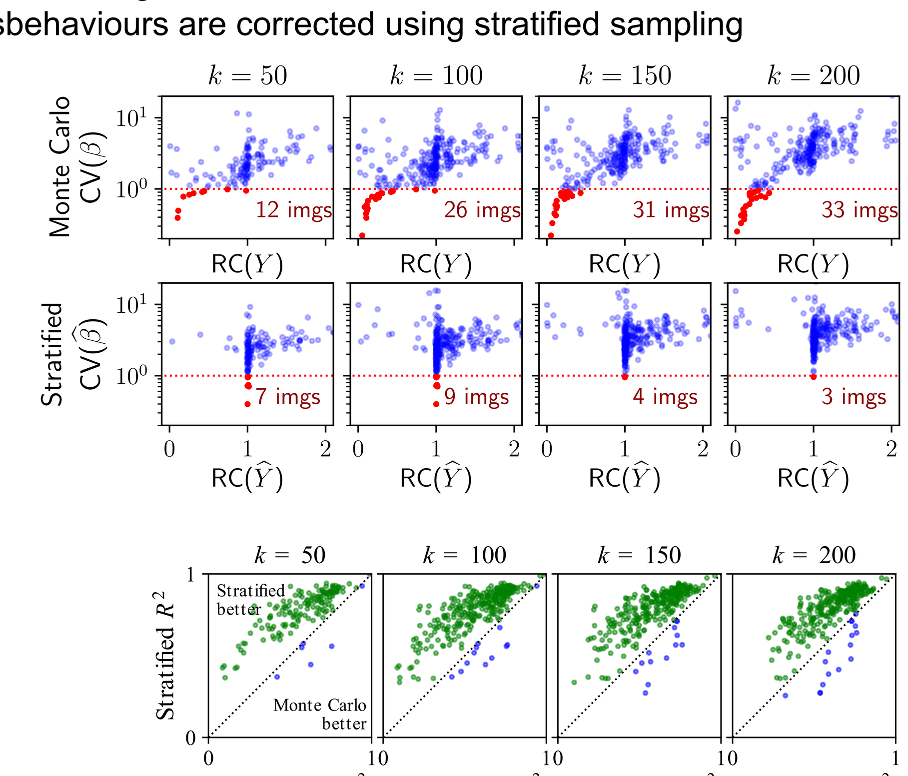
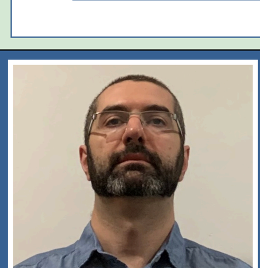


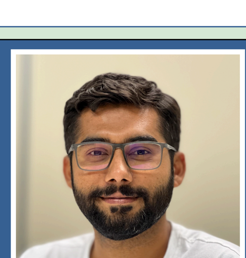
Image		Monte Carlo sampling (A) $k=50$ $CV(\beta) \mid RC(Y)$	Monte Carlo sampling (B) $k=200$ $CV(\beta) \mid RC(Y)$	Stratified sampling (C) $k=50$ $CV(\widehat{\beta}) \mid RC(\widehat{Y})$	Stratified sampling (D) $k=200$ $CV(\widehat{\beta}) \mid RC(\widehat{Y})$
Low CV Values	orange 71.3%	0.387 | 0.0833	0.178 | 0.0217	2.12 | 1.0	3.54 | 1.01
	wardrobe 31.4%	0.135 | 0.0182	0.164 | 0.00908	1.72 | 1.0	3.04 | 0.993
	milk_can 76.2%	0.967 | 0.357	0.571 | 0.145	3.04 | 1.14	4.71 | 1.14
	lynx 50.4%	1.57 | 1.19	0.464 | 0.137	3.05 | 1.71	4.79 | 1.58
	ringneck_snake 36.2%	1.10 | 0.365	0.221 | 0.0238	3.45 | 1.3	5.11 | 1.19
High CV Values	chickadee 99.8%	4.05 | 0.999	5.73 | 0.996	3.74 | 1.0	4.05 | 1.0
	polecat 46.9%	7.07 | 1.86	5.76 | 1.51	6.54 | 1.79	5.76 | 1.46

Conclusions

- Reformulation of LIME Image sampling strategy (not restricted to image data) for stratified sampling.
- Drawing lessons from the Shapley theory.
- Empirical evaluation shows that Monte Carlo undersampling is not rare, and stratified sampling provides practical improvements, at no additional cost.

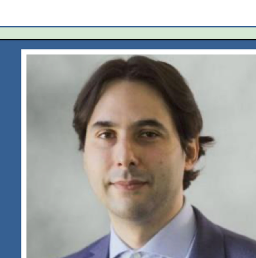


Elvio G. Amparore[1]
elviogilberto.amparore@unito.it



Muhammad Rashid[1]
muhammad.rashid@unito.it

Enrico Ferrari[2]
enrico.ferrari@rulex.ai



Damiano Verda[2]
damiano.verda@rulex.ai

Source code:

https://github.com/rashidrao-pk/lime_stratified
https://github.com/rashidrao-pk/lime-stratified-examples