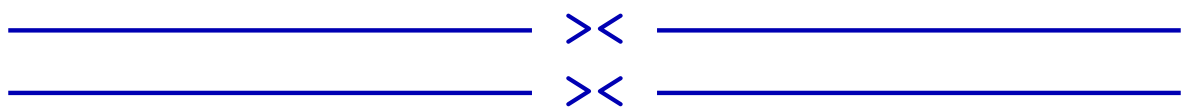
Data Shapley: Using Cooperative Game Theory to Detect Bad Training Data in Machine Learning

Abdul Basit

LUMS

April 20, 2026



ECON 3302: Introduction to Cooperative Game Theory

Presentation Outline

The Problem: Not All Training Data is Equal

Machine learning models are only as good as their training data.

Data Point Type	Effect on Model
Clean, representative	Improves accuracy
Mislabeled / corrupted	Actively hurts accuracy
Near-duplicate / redundant	No effect

Motivating Example (Medical Diagnosis):

- A hospital trains a cancer classifier on 1,000 patient records
- 200 records are mislabeled due to data entry errors
- How do you find **which records** are harmful, without checking all 1,000 manually?
- Naive fix: remove data randomly $\Rightarrow$ throws away good data too

Core Insight: A fair scoring method must capture each point's marginal contribution to the model, exactly what the Shapley value does.

Research Question

Central Question

Can the Shapley value assign a fair importance score to every training data point, such that mislabeled / corrupted data receives low scores while clean, informative data receives high scores, enabling automatic detection of bad data?

Three Sub-Questions (Hypotheses):

1. **Do corrupted data points receive negative Shapley values?**
   Points that hurt the model should score ≤ 0.
2. **Does removing low-Shapley data improve model accuracy?**
   Compare against random removal and Leave-One-Out as baselines.
3. **Does detection change as corruption rate increases?**
   Test at 10%, 20%, 30%, 40% corruption levels.

Our Approach: Apply the Data Shapley framework to the UCI Breast Cancer dataset with controlled label corruption, and run three additional extensions testing robustness, convergence, and generalization.

UCI Breast Cancer Wisconsin (Diagnostic) Dataset

Dataset at a Glance:

Property	Value
Total samples	569
Features per sample	30 (all real-valued)
Classes	2: Malignant (0) | Benign (1)
Class balance	212 malignant (37.3%) / 357 benign (62.7%)
Source	Fine needle aspirate (FNA) of breast mass image
Train split (80%, `rs=42`)	**455 samples**
Test split (20%)	**114 samples**

What the 30 features represent: Each sample is a digitized image of a cell nucleus. Ten geometric properties are measured three ways (mean, standard error, worst):

- **Radius**, **Texture**, **Perimeter**, **Area**, **Smoothness**, **Compactness**
- **Concavity**, **Concave Points**, **Symmetry**, **Fractal Dimension**

10 properties $\times$ 3 statistics (mean, SE, worst) = 30 features total

Dataset: Sample Data Points & Label Corruption

First 5 Features of 6 Representative Samples:

ID	Radius	Texture	Perimeter	Area	Smoothness	True Label
0	17.99	10.38	122.80	1001.0	0.1184	Malignant (0)
1	20.57	17.77	132.90	1326.0	0.0847	Malignant (0)
2	19.69	21.25	130.00	1203.0	0.1096	Malignant (0)
3	11.42	20.38	77.58	386.1	0.1425	Malignant (0)
5	12.45	15.70	82.57	477.1	0.1278	Benign (1)
6	18.25	19.98	119.60	1040.0	0.0946	Benign (1)

Malignant tumors tend to have larger radii, perimeters, and areas. However, the classes are **not linearly separable** on any single feature.

How corruption is introduced (our experiment):

- Randomly select $\lfloor 0.20 \times 455 \rfloor = \mathbf{91}$ training point indices
- Flip their binary label: $0 \to 1$ or $1 \to 0$
- *Features remain unchanged,* only the diagnostic label is wrong
- This mimics real-world data entry errors, annotation mistakes, or sensor failures

Mapping Training Data to a Cooperative Game

The Formal Setup (Coalitional Game with TU):

- **Players N:** Every individual training data point $\{(x_1, y_1), (x_2, y_2), \ldots, (x_{455}, y_{455})\}$
- **Coalition $S \subseteq N$:** Any subset of training points used to train the model
- **Value function $v(S)$:** The model's *test accuracy on 114 test points* when trained **only** on S

Concrete Example (5 points for illustration):

Coalition S	**Points Included**	$v(S)$ = Test Accuracy
$\{p_1\}$	1 clean point	61%
$\{p_1, p_2\}$	2 clean points	67%
$\{p_1, p_2, p_3\}$	2 clean + 1 corrupted	64% $\downarrow$
$\{p_1, p_2, p_4\}$	3 clean points	71% $\uparrow$

$v(\emptyset)$ = random guess accuracy; p_3 is corrupted — its inclusion hurts every coalition it joins.

Machine Learning Model: K-Nearest Neighbors (KNN)

What is KNN?

- A **non-parametric** classifier: stores the entire training set, no explicit training phase
- To classify a new point x^*: find its K nearest neighbors by Euclidean distance, take a majority vote of their labels
- Our experiment: $K = 5$, Euclidean distance, scikit-learn `KNeighborsClassifier`

Why KNN for Data Shapley?

Requirement	Why KNN satisfies it
Fast retraining	Adding/removing a point $\Rightarrow$ no fitting cost
Marginal sensitivity	Each new point can immediately shift neighbor vot
No convergence issues	Deterministic, no random initialization
Strong baseline	Achieves $\sim$89% accuracy on this dataset

Key Detail: TMC-Shapley evaluates $v(S)$ **1000 of times** per run. KNN makes each evaluation extremely fast, it simply uses whatever points are in S as the lookup table, with no gradient descent, no epochs.

Algorithm: Truncated Monte Carlo Shapley

The Core Challenge: Exact Shapley requires evaluating $v(S)$ for all $2^{455} \approx 10^{137}$ subsets, computationally impossible.

TMC-Shapley estimates via random permutations:

for $t = 1$ to T iterations **do**
  Sample random permutation π of all 455 training points
  $S \leftarrow \emptyset, \quad v_{\text{prev}} \leftarrow v(\emptyset)$
  for each point i in order π **do**
    $S \leftarrow S \cup \{i\}$
    $v_{\text{curr}} \leftarrow v(S)$ (train KNN on S, evaluate on 114 test pts)
    Record: $\delta_i^{(t)} \leftarrow v_{\text{curr}} - v_{\text{prev}}$
    if $|v_{\text{curr}} - v(\text{full set})| < \epsilon$ **then truncate** (skip remaining in π)
    end if
    $v_{\text{prev}} \leftarrow v_{\text{curr}}$
  end for
end for
$\hat{\phi}_i \leftarrow \frac{1}{T} \sum_{t=1}^{T} \delta_i^{(t)}$ (final Shapley estimate)

Experiment Parameters:

- **Main run:** $T = 1{,}500$ iterations (`save_every=100`, 15 checkpoints); error $\epsilon = 0.05$

Method: The Shapley Value (Chapter 18)

Core Idea: For each training point i, ask: across *every possible ordering* in which data points could be added to the training set, how much does point i improve the model when it joins?

The Formula:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(n - |S| - 1)!}{n!} \Big[v(S \cup \{i\}) - v(S) \Big]$$

- $v(S \cup \{i\}) - v(S)$ = point i's **marginal contribution** to accuracy when added to coalition S
- If p_i is mislabeled: $v(S \cup \{i\}) < v(S)$ for most $S \Rightarrow \phi_i < 0$
- The weight $\frac{|S|!(n-|S|-1)!}{n!}$ averages this contribution uniformly over all orderings

Computation Challenge: 2^n coalitions $\Longrightarrow$ infeasible exactly for 2^{455}.

Solution: Monte Carlo sampling: randomly sample permutations and average marginal contributions. Converges in practice with $\sim$1,500 samples.

Method: The Shapley Value — Four Axioms (Chapter 18)

The Four Shapley Axioms and what they mean for data:

Axiom	Meaning for Data
Efficiency	Total credit sums to overall model accuracy gain
Symmetry	Two data points with equal contributions get equal scores
Dummy Player	A point that never changes accuracy gets score $= 0$
Additivity	Scores combine correctly across independent tasks

Why naive alternatives fail:

- **Leave-One-Out (LOO):** Only checks "what if I remove just this one point?" ignores interactions between data points; violates Symmetry in practice
- **Data frequency / class weight:** Treats all points in a class equally; violates Dummy Player (redundant duplicates get positive weight despite adding no new information)

The Shapley value is the unique function satisfying all four axioms simultaneously.

Experimental Setup

Step-by-Step Pipeline:

1. **Load dataset:** 569 samples, 30 features, stratified 80/20 split (`random_state=42`)
2. **Corrupt labels:** Randomly select 91 training points and flip binary label ($0 \leftrightarrow 1$)
3. **Run TMC-Shapley:** 1,500 iterations with KNN (K=5), ϵ=0.05; also compute LOO values
4. **Evaluate three removal strategies:** Shapley, random, LOO

Parameter	**Value**	**Reason**
Corruption rate (main)	20% (91 pts)	Realistic noise level
Removal fraction	20% (91 pts)	Matches corruption count
KNN K	5	Scikit-learn default
TMC iterations (main)	1,500	Convergence verified
Error threshold ϵ	0.05	5% stability criterion
Random seed (split)	42	Reproducibility

Hypotheses

H1: Corrupted Points Get Negative Scores:

Mislabeled training points will receive **negative** Shapley values, because their inclusion consistently reduces model accuracy across coalitions.

H2: Shapley-based Removal Outperforms Baselines:

Removing the bottom data points by Shapley value will **improve model accuracy more** than removing the same number randomly, and significantly more than the Leave-One-Out baseline.

H3: Detection Degrades at High Corruption:

Shapley-based detection will remain effective up to a certain corruption level, but **break down beyond that level**, because at majority corruption the "clean" coalition becomes the minority.

Result: H1: Shapley Value Distribution

Experiment: 20% label corruption (91/455 training labels flipped), 1,500 TMC-Shapley iterations.

Data Group	Count	Average Shapley Value
Clean labels	364	$+0.0021$
Corrupted labels	91	-0.0071

Key Findings:

- Corrupted points receive Shapley values 3.4× **further from zero**
- The corrupted distribution is shifted clearly into negative territory
- Clean data clusters around small positive values
- A few clean points near the decision boundary also receive small negative values, these are ambiguous points, not misdetections

Verdict: H1 CONFIRMED

Corrupted points consistently receive negative Shapley values, clearly separated from clean data.

Result: H2: Removal Strategy Comparison

Setup: Remove the bottom 20% of training points (91 points) by each strategy; retrain and evaluate on 114 test points.

Strategy	Test Acc	Improvement	Detection Rate
Baseline (no removal)	88.60%	—	—
Random Removal	87.72%	−0.88%	21.98%
LOO Removal	96.49%	+7.89%	29.67%
Shapley Removal	**96.49%**	**+7.89%**	**82.42%**

Interpretation:

- Random removal *hurts* accuracy, removes more good data than bad
- Shapley removal identifies **82.4%** of corrupted points using only a 20% removal budget
- LOO matches Shapley's *accuracy improvement* (lucky on accuracy) but finds only 29.7% of corrupted points, not a reliable detector

Verdict: H2 CONFIRMED

Shapley-based removal vastly outperforms random removal.

Result: H3: Detection Across Corruption Rates

Setup: Test 10%, 20%, 30%, 40% corruption; 20 TMC iterations per rate; remove 20% of points by each strategy.

Corrup.	Baseline Acc.	Shapley Det.	Random Det.	LOO Det.
10%	88.60%	**47.25%**	7.69%	20.88%
20%	85.96%	**79.12%**	17.58%	30.77%
30%	77.19%	**92.31%**	32.97%	45.05%
40%	72.81%	**96.70%**	43.96%	63.74%

Surprising Finding: Detection improves as corruption increases!

- At 40% corruption, Shapley detects **96.7%** of corrupted points
- **Why?** More corrupted points ⇒ each corrupted point has a *more consistent* negative marginal contribution across permutations ⇒ a clearer, stronger signal

Verdict: H3 REJECTED (pleasantly)

Shapley-based detection does **not** break down at high corruption, it becomes **more reliable,** not less.

Summary of All Results

Hypothesis / Extension	Key Finding	Verdict
H1: Corrupted $\Rightarrow$ negative ϕ_i	-0.0071 vs. $+0.0021$ avg	**Confirmed**
H2: Shapley removal $>$ baselines	82.4% vs. 22.0% detection	**Confirmed**
H3: Detection degrades	Improves to 96.7% at 40%	**Rejected**

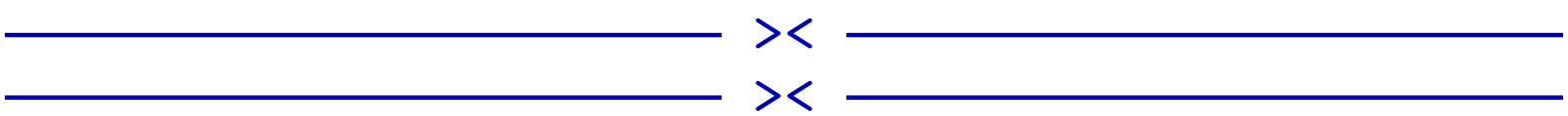
What This Project Demonstrates:

- The **Shapley value** provides a principled, axiomatically justified score for every training data point
- It reliably identifies mislabeled data across **corruption rates, random splits**
- Practical application: **automatic data cleaning** for medical, legal, and financial datasets before model training

Conclusions

Core Takeaways:

1. **Shapley values work as a data quality score.** Corrupted points receive negative values; clean points receive positive values. The separation is clear and statistically robust (5-fold CV, $\pm 1.3\%$).
2. **Removal of low-value data recovers accuracy.** Removing 20% of training points (guided by Shapley) pushes test accuracy from 88.6% to 96.5%, a gain of $+7.9$ percentage points.
3. **Cooperative game theory provides something unique.** The Shapley value considers *all possible coalitions* of data points, capturing interactions that single-point methods (LOO) miss entirely.



ECON 3302: Introduction to Cooperative Game Theory