

# Bayes Factor: Model-Test Compatibility using the Empirical-Bayesian Diagnosis and MLOps Monitoring without Feature Access

Fabricio Echeverria<sup>1</sup>, Marcelo Leon<sup>2,3,\*</sup>, Narcisa Vasquez<sup>4</sup>, Veronica Limones<sup>4</sup> and María-Fé Jarrín<sup>4</sup>

<sup>1</sup>Corporación Universitaria Asturias, Colombia

<sup>2</sup>Universidad ECOTEC, Samborondón, Ecuador

<sup>3</sup>Universidad Continental, Lima, Perú

<sup>4</sup>Universidad Estatal Península de Santa Elena, La Libertad, Ecuador

## Abstract

This article allows us to evaluate whether the behavior of a classification model on the test set is consistent with its behavior in training, a verification problem rarely addressed quantitatively. This work proposes the Bayes Factor (BF10) as a measure of model-test compatibility. Seven controlled experiments are presented on eight datasets ( $n = 150 - 284, 807$ ) covering very different scales, feature types, and imbalance levels: (A) detection of four split types using the Wilcoxon test on  $N = 20$  replicates; (B) size sufficiency analysis; (C) noise degradation in labels; (D) quantitative comparison of the BF against the KS-test, MMD, log-loss, softmax entropy, and BBS using AUC-ROC; (E) class-based diagnosis; (F) sensitivity to the  $\lambda$  factor; and (G) validation with real time shift in Credit Card Fraud. The BF is a bidirectional diagnostic: it can increase (artificially easy test) or decrease (shift or bias difficult), so the correct evaluation metric is the absolute deviation from the reference BF. Under this metric, the BF achieves AUC=0.91–0.95 (LR/RF), competitive with prediction-based baselines (log-loss AUC=0.953–0.957; BBS AUC=0.887–0.992) while operating solely on  $(k, n)$  without calibrated probabilities or auxiliary discriminators. In the actual time shift (Exp G), the BF drops by -73%, offering the clearest drift signal among all detectors. BF  $d = \log_{10} BF/n$  test is also proposed as a normalized version that eliminates dependence on test size. These properties make the BF a practical metric for MLOps and monitoring production models: it operates without access to test features, is computable on prediction batches of any size (thanks to BF  $d$ ), and the  $\Delta H$  metric acts as a batch sufficiency detector before issuing an alert.

## Keywords

Bayes factor, Model-test compatibility, Predictive consistency, Distribution shift, Conjugate Beta-Binomial, Shift detection, MLOps

## 1. Introduction

In the standard supervised learning cycle, the test set acts as an impartial referee: its examples are assumed to come from the same distribution as the training set, and therefore, the accuracy on the test set is an unbiased estimator of the model's generalization ability. This assumption—often called the IID assumption—is frequently violated in practice.

1. **Distribution shift:** Train and test are obtained from different time periods, geographical regions, or populations.
2. **Class bias:** The fraction of classes in the test differs systematically from that in the training.
3. **Label contamination:** Errors in annotation in the test introduce noise that artificially degrades accuracy.

ICAIW 2026: Workshops at the 9th International Conference on Applied Informatics 2026, October 13–17, 2026, Orlando, United States

\*Corresponding author.

✉ pedro.echeverria@asturias.edu.co (F. Echeverria); marceloleon11@hotmail.com (M. Leon); nvasquez@upse.edu.ec (N. Vasquez); jlmones@upse.edu.ec (V. Limones); mafejarrin.99@gmail.com (M. Jarrín)

ORCID 0000-0002-4921-1111 (F. Echeverria); 0000-0001-6303-6615 (M. Leon); 0000-0002-7065-4216 (N. Vasquez); 0009-0003-0468-3873 (V. Limones); 0009-0004-5439-4305 (M. Jarrín)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

4. **Insufficient size:** A very small test set produces accuracy estimates with high variance, with no discriminative value.
5. **Silent degradation in production:** A deployed model receives batches of new data whose distribution may gradually deviate from the training data. Without a quantitative mechanism for comparing the observed behavior in production with the learned behavior, the degradation goes unnoticed until the business impact becomes apparent.

The practical problem is that these violations are difficult to detect with point accuracy: low accuracy can be due to a weak model or an inadequate test. Without a quantitative diagnosis of the compatibility between training and testing, it is impossible to decouple the two causes.

The context of MLOps amplifies this problem: models are deployed in automated pipelines where production data is not always available with full features, and monitoring systems must operate with the minimum amount of information possible [1]. Most existing monitoring tools (Evidently AI, WhyLabs, Seldon Alibi) require access to test features to calculate shift metrics. This makes them inapplicable in black-box scenarios, where the model is exposed as an API, and only its predictions are available.

This work proposes to formalize that compatibility using the Bayes Factor (BF). The central idea is to interpret training as evidence that builds a prior distribution over the model’s accuracy rate (Empirical Bayes), and the test as evidence that updates that prior distribution. Under this framework, BF quantifies how many times more likely the test’s accuracy rate is under the learned distribution than under a random classifier. A high BF implies compatibility (the test confirms what was learned); a low or negative BF indicates that the test is inconsistent with the training—a sign of inadequacy.

The main contributions of the work are:

- Propose the BF as a measure of model-test compatibility (not of absolute test quality), with dual interpretation: inconsistency due to test shift/bias or overfitting of the model.
- Validate the diagnosis in seven controlled experiments with the Wilcoxon test for statistical significance ( $N = 20$  repetitions) on eight datasets ( $n = 150 - 284, 807$ ), plus validation with real-time shift.
- Comparing BF against KS-test, MMD, log-loss, softmax entropy, and BBSD [2] using AUC-ROC, showing that BF is competitive (AUC0, 91–0, 95) using only  $(k, n)$ -without calibrated probabilities or auxiliary discriminators.
- Propose BF  $d = \log_{10} BF / n$  test as a normalized version independent of test size.
- Identify  $\Delta H$  as an indicator of size sufficiency.
- Extend the diagnosis to the individual class level by using BF per class (Experiment E), locating which specific class generates the model-test incompatibility.
- Validate with real-time shift in Credit Card Fraud (Exp G): BF falls by 73% in the time split, offering the clearest drift signal among all detectors.
- Propose a **production monitoring protocol (MLOps)** based on three steps: (1) calculate BF on the batch of predictions; (2) verify batch sufficiency with  $\Delta H > 0$ ; (3) issue an alert if  $|BF d - BF d_{baseline}|$  exceeds a threshold. This protocol operates without access to the test features.
- Provide reproducible open-source code in Python.

## 2. Related Work

### 2.1. Distribution shift detection

Detecting changes in data distribution is an actively studied problem in machine learning. Existing methods are grouped into two families according to the space in which they operate.

**Feature-based methods (X-space):** The Kolmogorov-Smirnov test (KS) contrasts the empirical distribution of each feature between the train and test sets, reporting the maximum discrepancy as a statistic [3]. The Maximum Mean Discrepancy (MMD) [4] provides a global measure in the reproducible

space of an RBF kernel, capturing higher-order differences. Both methods require explicit access to the features of the test set. Methods such as KSWIN [5] and the Drift Detector Method (DDM) [6] extend these ideas to online data streams.

**Prediction-based methods (space and  $\theta$ ):** A second family evaluates the distribution of predictions or losses instead of features. The method by Lipton et al. [2] detects label shift by comparing prediction distributions between train and test. Concept drift detectors [7] monitor the error rate in streams without needing the original features. Our work falls into this family: the BF compares the predictive distribution of the model on the test (encoded in  $k$  test,  $n$  test) with the distribution learned in training (prior Beta-Binomial).

## 2.2. Bayes Factor in Machine Learning

Bayesian factorization (BF) has been extensively studied as a hypothesis testing tool in statistics [8, 9, 10] and quantitative psychology [11, 12]. Its direct application to diagnosing machine learning splits is, to our knowledge, novel. Related work includes the use of BF to compare classifiers [13] and the Bayesian evaluation of variable coefficient algorithms [14], but none reinterprets it as a diagnostic tool for test set adequacy.

## 2.3. Dataset Shift

The theoretical framework of the dataset shift [15] distinguishes between covariate shift (change in  $P(X)$ ), label shift (change in  $P(Y)$ ), and concept drift (change in  $P(Y|X)$ ). Our method detects the consequences of any of these shifts in the prediction space, without requiring knowledge of which type of shift occurred.

# 3. Theoretical Framework

## 3.1. The Beta-Binomial Conjugate Model

Let  $\theta \in (0, 1)$  be the probability that the model correctly classifies an arbitrary example. We model  $\theta$  as a random variable with Beta distribution and the observed evidence as Binomial:

$$\theta \sim \text{Beta}(\alpha, \beta), k|\theta, n \sim \text{Binomial}(n, \theta) \quad (1)$$

Beta-binomial conjugation guarantees an analytically closed posterior:

$$\theta | k, n \sim \text{Beta}(\alpha + k, \beta + n - k) \quad (2)$$

## 3.2. Prior Empirical from Training (Empirical Bayes)

The Empirical Bayes strategy [16, 17] consists of estimating the prior hyperparameters from the observed data before new evidence arrives.

Here, the training set is that source: given  $k$  train hits over  $n$  train examples, we construct the prior with Laplace smoothing (to avoid degeneracy at the extremes):

$$\alpha_0 = k_{\text{train}} + 1, \beta_0 = (n_{\text{train}} - k_{\text{train}}) + 1 \quad (3)$$

The mean of this prior is  $\mathbb{E}[\theta] \approx k_{\text{train}}/n_{\text{train}}$ , that is, the training accuracy. The prior does not represent beliefs prior to the observation of any data; it represents what the model learned: the probability distribution over  $\theta$  compatible with its behavior in training.

This reinterpretation is key: the prior is not external information to the problem, but rather the encoding of the model's learning. The question that the BF answers is then: *Is the performance observed in the test consistent with what the model learned?*

### 3.3. Subsequent Update with Test

After observing  $k_{test}$  successes on  $n$  test samples from the test set:

$$\alpha_{post} = \alpha_0 + k_{test}, \beta_{post} = \beta_0 + n_{test} - k_{test} \quad (4)$$

The 95% credibility interval (95% CI) is derived from the 2.5% and 97.5% Beta ( $\alpha_{post}, \beta_{post}$ ).

### 3.4. Bayes Factor as a Compatibility Diagnostic

We define the compatibility hypotheses:

- $H_1 : \theta \sim \text{Beta}(\alpha_0, \beta_0)$  (success rate distributed according to what was learned in training).
- $H_0 : \theta = p_0 = 1/C$  (random classifier over  $C$  classes - without training information).

$$BF_{10} = P(k_{test}|n_{test}, H_1) / P(k_{test}|n_{test}, H_0) \quad (5)$$

where the marginal likelihood under  $H_1$  is:

$$P(k|n, \alpha, \beta) = nB(\alpha + k, \beta + n - k) / kB(\alpha, \beta) \quad (6)$$

and under  $H_0 : P(k|n, p_0) = n \binom{n}{k} p_0^k (1 - p_0)^{n-k}$ . For numerical stability, it is computed on a logarithmic scale:  $\log BF_{10} = \log P(k|H_1) - \log P(k|H_0)$ .

Under this interpretation,  $BF_{10} > 1$  indicates that test performance is more likely to be based on the learned distribution than on chance; that is, the test is consistent with training. The interpretation follows Jeffreys' scale [8] (Table 1).

**Table 1**

Jeffreys Scale for the interpretation of the Bayes Factor

| $BF_{10}$ | Evidence for $H_1$ (train-test compatibility) |
|-----------|-----------------------------------------------|
| $> 100$   | Extreme: highly compatible test               |
| 30 - 100  | Very strong                                   |
| 10 - 30   | Strong                                        |
| 3 - 10    | Moderate                                      |
| 1 - 3     | Anecdotal                                     |
| $< 1$     | Incompatible: test not suitable               |

### 3.5. Information Gain and Size Sufficiency

The differential entropy of Beta ( $\alpha, \beta$ ) is:

$$H(\alpha, \beta) = \ln B(\alpha, \beta) - (\alpha - 1)\psi(\alpha) - (\beta - 1)\psi(\beta) + (\alpha + \beta - 2)\psi(\alpha + \beta) \quad (7)$$

where  $\psi$  is the digamma function. The information gain provided by the test is defined as:

$$\Delta H = H(\text{Prior}) - H(\text{Post}) \quad (8)$$

$\Delta H > 0$ : the test reduced the uncertainty about  $\theta$  -the test provides information.  $\Delta H \approx 0$ : the test does not materially modify the prior-insufficient to resolve the uncertainty.  $\Delta H < 0$ : the posterior is more dispersed than the prior, which occurs when training accuracy is almost perfect, and the incompatibility of the test forces the posterior towards more uncertain intermediate values.

This metric acts as an indicator of sufficiency: a test that is too small produces  $\Delta H \approx 0$  because  $n_{tests}$  are insufficient to update a prior built on thousands of training samples.

## 4. Methodology

### 4.1. Datasets

Eight public datasets are used that cover a wide range of scale, number of classes, and feature types (Table 2). They were chosen for their public availability, diversity of conditions, and representativeness of real-world scenarios in MLOps.

**Table 2**

Datasets used in the experiments. All are public and available from their indicated sources.

| Dataset           | Characteristics                                | $n$    | $C$ | $p_0 = 1/C$ |
|-------------------|------------------------------------------------|--------|-----|-------------|
| Iris              | 4 numerical features, classic reference        | 150    | 3   | 0.333       |
| Penguins          | 4 numeric features, successor to Iris          | 333    | 3   | 0.333       |
| Haberman's        | 3 features, size limit case                    | 306    | 2   | 0.500       |
| Dry Bean          | 16 numerical features, 7 classes (UCI #602)    | 13611  | 7   | 0.143       |
| Adult             | 14 mixed features, revenue forecast            | 48598  | 2   | 0.500       |
| Bank Marketing    | 20 mixed features, telemarketing (UCI #222)    | 45211  | 2   | 0.500       |
| Credit Card Fraud | 30 features (PCA), financial fraud, Kaggle     | 284807 | 2   | 0.500       |
| Fashion-MNIST     | 784 image features, 10 classes (subsample 10k) | 10000  | 10  | 0.100       |

### 4.2. Models

Two representative models are evaluated:

- **Logistic Regression (LR)**: linear model, max\_iter =500.
- **Random Forest (RF)**: 100 trees, random\_state =42.

The evaluation protocol uses  $N = 20$  repetitions per condition to average the random variability of the splits.

### 4.3. Split Types (Experiment A)

**S1. Stratified**: train\_test\_split with stratify=y. Identical class distribution in train and test. Appropriate split reference.

**S2. Random**: without stratification. Small random deviations in the class distribution. Moderately adequate.

**S3. Feature shift**: the data is sorted by the feature with the highest variance, and the initial segment is assigned to the train and the final segment to the test. This produces a controlled distribution shift: the train and test have non-overlapping feature distributions.

**S4. Class bias**: 90% of the examples in class 0 are assigned to the test, with the remainder normally distributed. The test is overrepresented by one class.

### 4.4. Bayesian Evaluation Protocol

For each split  $(X_{tr}, X_{te}, y_{tr}, y_{te})$ :

1. Train the model on  $(X_{tr}, y_{tr})$ .
2. Calculate  $(k_{train}, n_{train})$  and differentiate Beta( $\alpha_0, \beta_0$ ).
3. Evaluate on  $(X_{te}, y_{te})$ ; obtain  $(k_{test}, n_{test})$ .
4. Computar posterior,  $BF_{10}$ ,  $\Delta H$  e IC95 %.

The metrics are averaged over  $N = 20$  repetitions. The code is organized into three Python modules: bayesian\_cv.py (pure functions), datasets.py (loaders), and adequacy\_experiment.py (runner).

## 5. Results

Due to the length of the results, the tables from each experiment are attached at the following link: <https://n9.cl/8qwo4>. There, you can find the results by referring to the sheet names, which are labeled with the names of each experiment, for example, Experiment A.

### 5.1. Experiment A: Detection of Distribution Shift and Class Bias

The table at the link shows the results of Experiment A (averages over 20 repetitions).

**Finding A1 – Distribution Shift Detection:** The clearest case is Penguins with RF: the stratified split produces  $\log_{10} BF = 27.56$  (extreme evidence of compatibility), but the split due to shift drops to  $\log_{10} BF = -2.94$  (incompatibility). The test accuracy falls from 97.5% to 73.1%, which the BF amplifies: the prior learned in training ( $accuracy \approx 1.0$ ) is completely incompatible with that 73%. Similarly, in Dry Bean, the feature shift reduces the  $\log_{10} BF$  from 1.345 to 509 for LR (a 62% drop), and from 1.690 to 628 for RF (63%). In Bank Marketing, the shift is even more drastic: the RF BF drops from +950 (stratified) to -1.345 (shift), crossing the indifference threshold. In Fashion-MNIST, the shift reduces the BF of LR from 1,228 to 1,161 (-5.5 %) and that of RF from 1,135 to 1,021 (-10 %): a moderate drop consistent with the nature of the highest variance feature in a 784-dimensional space.

**Finding A2 - Class Bias: Asymmetric Effect:** The effect of class bias is asymmetric depending on which class is overrepresented. In Haberman RF, the BF falls from -7.63 to -23.80 (overrepresented difficult class): the test contains predictions that the overfitted model cannot replicate. In Adult, the bias toward the difficult class produces  $BF = -609.70$  for LR and -487.76 for RF, compelling evidence of incompatibility. In contrast, in Iris, Penguins, and Bank Marketing, the BF increases with class bias (overrepresented easy class), reaching +3,960 in Bank Marketing LR: The artificially easy test generates spurious evidence of compatibility. This finding reveals a second type of diagnosis: an inflated BF compared to the stratified one may indicate an artificially easy test (bias towards single classes).

### 5.2. Experiment B: Size Sufficiency

The table at the link shows the behavior of BF and  $\Delta H$  when varying the test percentage from 5% to 50% (*View table at the link*).

The BF grows essentially linearly on a logarithmic scale, consistent with the asymptotic relationship  $\log BF \propto nD KL(\theta||p_0)$ . The  $\Delta H$  metric is more informative for detecting the sufficiency threshold: for Iris,  $\Delta H$  increases from 0.037 (5%) to 0.139 (20%) and does not grow substantially beyond 30%. For large datasets like Dry Bean, the BF continues to grow, but the differential gain per additional percentage point decreases, suggesting that the marginal information of the test is exhausted.

### 5.3. Experiment C: Label Noise Degradation

The BF falls monotonically with noise across all datasets and models. The crossover point  $\log_{10} BF = 0$  (indifference boundary) occurs around 25-30% noise for Iris and Penguins: the model declares incompatibility with the test when more than a quarter of its labels are incorrect. For Dry Bean, given its larger n-test size, the threshold shifts to 50%. This behavior establishes the BF as a label contamination detector with sensitivity adjustable to the test size (*View table at the link*).

### 5.4. Experiment D: BF vs KS-test vs MMD

datasets. The fundamental distinction is operational: KS and MMD require access to the test features ( $X_{te}$ ), while BF operates only on the predictions ( $k_{test}, n_{test}$ ).

**Note on the choice of MMD kernel** An RBF kernel with bandwidth  $\gamma = 1/d$  (where  $d$  is the number of features) was used, which is the default value of `rbf_kernel` in `scikit-learn` and the standard choice in the literature. To verify that the results are not artifacts of the hyperparameter, an additional search was performed with  $\gamma \in \{0, 1, 0.5, 1, 0.2, 0.05, 0\}/d$ : in all cases, the AUC of the MMD in the

class bias scenario remained in the range 0.50–0.57, confirming that the MMD’s failure to detect class bias is structural, not an artifact of the kernel choice (*View table at the link*).

KS and MMD measure shift in feature space; BF measures compatibility in prediction space. Underlined: MMD is lower than that of the stratified split (does not detect inadequacy). Bold: clear detection of the shift.

The results reveal complementary and, in part, divergent detection profiles:

**Feature shift:** KS = 1.0 across all datasets (perfect detection in feature space). However, in Penguins, the MMD of the shift split (0.0197) is virtually identical to that of the stratified split (0.0187): the shift created by ordering by the feature with the highest variance is not captured by the global RBF kernel. In contrast, the RF BF drops from +27.56 to -2.94, indicating that the ordered test is incompatible with what was learned during training.

**Class bias:** This is where feature-based methods show the greatest weakness. In Haberman, the MMD of the class-biased split (0.0163) is lower than that of the stratified split (0.0201): the overrepresentation of one class reduces the overall feature dispersion without creating a detectable kernel shift. The BF of the RF, however, drops from -7.63 to -23.80, capturing the prediction incompatibility. In class-biased Penguins, the MMD also falls below that of the stratified split (0.0122 vs. 0.0187), while the BF increases (+63.05), indicating an artificially easy test.

The operational difference has direct practical implications: in scenarios where the test comes from an external source (production API, monitoring system) and only the model’s predictions are available, KS and MMD are inapplicable. BF, built on  $(k_{test}, n_{test})$ , operates in that context without modification.

**Prediction-based baselines (R3/R13):** Three additional detectors were included that, like the BF, do not require access to the test features  $X_{te}$ : (i) log-loss deviation ( $|L_{te} - L_{ref}|$ , requires predict\_proba), (ii) mean entropy deviation ( $|H_{te} - H_{ref}|$ ), and (iii) BBSD: a logistic discriminator trained to separate the predict\_proba( $X_{tr}$ ) vs predict\_proba( $X_{te}$ ) distributions, with AUC as the shift score [2]. Table 3 shows the AUC-ROC of all detectors.

**Table 3**

Experiment D - Binary Detection AUC-ROC (adequate vs inadequate).  $BF_{dev}$ ,  $LogLoss_{dev}$ ,  $Entropy_{dev}$ , BBSD: predictions only. KS, MMD: require test features. Bold: best of the prediction-only detectors.

| Model | $BF_{dev}$ | <b>KS</b> | <b>MMD</b> | $LogLoss_{dev}$ | $Entropy_{dev}$ | <b>BBSD</b> |
|-------|------------|-----------|------------|-----------------|-----------------|-------------|
| LR    | 0.910      | 0.949     | 0.582      | 0.957           | 0.973           | 0.992       |
| RF    | 0.949      | 0.949     | 0.582      | 0.953           | 0.996           | 0.887       |

Prediction-based detectors show high AUC (0.887–0.996) when calibrated probabilities are available. However, their applicability is more restrictive than BF: log-loss and entropy require predict\_proba per sample (not available in non-probabilistic models or APIs that return only the predicted class); BBSD additionally requires retraining a discriminator. BF, with AUC 0.910-0.949, operates only on the  $(k, n)$  pair -the minimum sufficient statistic for accuracy - and is directly applicable in monitoring scenarios where only batch counts of correct predictions are available.

## 5.5. Experiment E: Class-Based Diagnosis (BF by Class)

Experiment E extends the Beta-Binomial model to the level of each class  $c$ . For each class, an independent prior Beta ( $\alpha$  0,  $c$ ,  $\beta$  0,  $c$ ) is constructed from the recall rate of that class in training, and BF  $c$  is calculated on the recall of class  $c$  in the test. This extension answers the question: which specific class does this incompatibility between the model and the test cause?

The table added in the link shows the most relevant results (Logistic Regression; means over  $N = 20$  repetitions) (*View table at the link*).

The results reveal three diagnostic patterns per class:

**Pattern E1 - Feature shift localization** In Dry Bean with LR, feature splitting produces  $BF C_4 = -36.3$  while  $BF C_2 = 987.9$ : the incompatibility is specifically concentrated in class 4, which is almost

absent from the test due to feature sorting. The overall BF of 509 (Exp A) averaged these heterogeneous contributions, obscuring the location of the problem.

**Pattern E2 – Class bias: localized inflation** The bias towards class 0 inflates BF  $C_0$  in isolation: Iris 4, 7  $\rightarrow$  20, 5, Penguins 12, 5  $\rightarrow$  47, 3, Haberman LR 9, 0  $\rightarrow$  40, 3. The other classes maintain values comparable to the stratified split. This diagnosis is impossible to obtain with the global BF.

**Pattern E3 - Localized overfitting in minority class** In Haberman RF, the stratified split produces BF  $C_1 = -6, 2$  (class 1 = 5-year survivors, minority class) while BF  $C_0 = 1, 0$ . This indicates that the RF overfitting operates specifically on the hard class, not the easy one.

## 5.6. Experiment F: Sensitivity to Factor $\lambda$ (Dominant Priority)

When the model overfits in training (acc train  $\approx 1$ ), the prior Beta ( $n_{train} + 1, 1$ ) becomes extremely concentrated at  $\theta \approx 1$ . Any test with moderate accuracy produces  $\log_{10} BF < 0$ , even for perfectly stratified splits-which can be misinterpreted as test incompatibility when the real source is model overfitting. The factor  $\lambda$  introduces a smoothing of the prior:

$$\alpha_0 = \lambda \cdot k_{train} + 1, \beta_0 = \lambda \cdot (n_{train} - k_{train}) + 1, \lambda \in (0, 1] \quad (9)$$

$\lambda = 1$  corresponds to the original behavior;  $\lambda < 1$  smooths the prior towards Beta (1, 1) (uniform). The table in the link shows the results for RF (most affected case) in three datasets representative of the effect (Haberman, Penguins, Dry Bean); the remaining five exhibit analogous behaviors. The stratified split should produce a positive BF; the shift, a lower BF (*View table at the link*).

The results reveal a trade-off between smoothing and sensitivity. In Haberman RF,  $\lambda = 0.25$  reduces the stratified BF from -7.6 to -2.5 but does not make it positive: the severe overfitting of the RF (acc train = 0.985) cannot be compensated for by  $\lambda$  alone, confirming that the negative BF in this case genuinely reflects the model’s overfitting. In Penguins,  $\lambda = 1.0$  detects the shift (-2.9 vs +27.6), but  $\lambda = 0.25$  loses that sensitivity (+4.3 vs +28.0, now indistinguishable). The value  $\lambda = 0.75$  maintains the detection (-1.2) with moderate smoothing.

**Practical recommendation**  $\lambda = 0.9$  is proposed as the default value: this produces minimal smoothing that does not affect detection capability under normal conditions but reduces the dominance of the prior in cases of extreme overfitting. To calibrate  $\lambda$  in production, it is suggested to find the value that maximizes the nominal 95% CI coverage in an internal hold-out of the training set, so that the prior remains calibrated.

## 5.7. Experiment G: Validation with Real Temporal Shift (Credit Card Fraud)

Experiment G validates BF in a real-world time-shift scenario: the Credit Card Fraud dataset contains 284,807 transactions ordered chronologically by the Time column. This scenario replicates the most common MLOps use case: a model trained on a historical period that is deployed in production and tested against data from a later period.

**Protocol.** 50,000 observations were sampled, preserving the temporal order. (Overrepresented frauds to maintain statistical signal). The time split assigns the initial 80% (chronological) to training and the final 20% to testing. The stratified split (reference) performs train\_test\_split (stratify=y, random\_state=42). Only Logistic Regression is used (RF is too expensive at 284k rows).

The fraud rate differs between periods: 0.0018 in the initial 80% vs. 0.0013 in the final 20%, which introduces a slight but real distributional shift in the distribution of the target variable (*see the table in the link*).

The BF falls from +1076.7 to +289.3 in the time split (-73%), indicating that the model’s behavior in the later period is considerably less compatible with learning in the earlier period. The test accuracy falls from 0.837 to 0.734, confirming a real degradation. The prediction-based detectors show contrasting sensitivities: BBSO hardly changes (0.498  $\rightarrow$  0.583), and the log-loss is practically identical (0.632 vs 0.648), while the BF provides the clearest signal of temporal drift with only the  $(k, n)$  pair.

This result illustrates the main advantage of BF in production: it quantifies compatibility degradation in distributional time shift scenarios using only batch hit counts - without the need for calibrated probabilities or additional discriminators.

## 6. Discussion

### 6.1. Model-Test Compatibility: Dual Interpretation

The central finding of this work is that the BF measures model-test compatibility, not the quality of the test in an absolute sense. This distinction is fundamental for a correct interpretation. A low BF has two interpretations:

- **Test inconsistent with training:** the test comes from a different distribution (shift), is biased towards difficult classes, or its labels are corrupted. In all these cases, the test is not an impartial arbiter of model learning.
- **Overfitted model:** the prior is excessively concentrated at  $\theta \approx 1$  (RF case with  $acc_{train} = 1, 0$ ) and any test with  $acc < 1, 0$  produces a negative BF, even if the test is perfectly representative. In this case, the BF is diagnostic of overfitting, not of test inadequacy.

This ambiguity is a strength, not a limitation: the BF detects inconsistencies in the evaluation pipeline regardless of their origin. To disambiguate, simply compare the BF of the overfitted model (RF) with that of a more regular model (LR): if the BF drops only for RF, the source is overfitting; if it drops for both models, the source is the test. Experiment A demonstrates precisely this pattern in Penguins (split by shift: LR maintains a high BF, RF collapses to -2.94) and Haberman (class bias: both models produce anomalous BFs).

### 6.2. Three Patterns of Predictive Inconsistency

**P1. Distribution shift: sharp drop in BF (model-dependent)** When the test comes from a region of the feature space not covered by the training, models that depend on that structure (RF) collapse, while more global models (LR) can be robust. BF captures this asymmetry: Penguins RF drops from +27.56 to -2.94 while LR remains at +31.67. Statistically significant difference ( $p < 0.001$ , Wilcoxon).

**P2. Class bias: Abnormally high or low BF** A test overrepresented by a difficult class produces very low BF (Haberman RF: -23.80 vs -7.63,  $p < 0.001$ ). A test overrepresented by an easy class produces inflated BF (Penguins LR: +63.05 vs +29.84,  $p < 0.001$ ). In both cases, the BF indicates that the test is not a balanced arbiter; the MMD, on the other hand, does not detect this bias (Exp D).

**P3. Label contamination: monotonic degradation of BF** BF drops linearly with the noise percentage and crosses the indifference boundary around 25-30% contamination (Exp. C). This behavior can be used as an operational threshold: if the observed BF is lower than the expected BF under a clean split, the test is contaminated.

### 6.3. Normalized BF: $BF = \log_{10} BF / n_{test}$

Given that  $\log BF_{10} \propto n_{test} \cdot D_{KL}(\theta^{test} || \theta^{train})$  asymptotically, the normalized version:

$$BF = d(\log_{10} BF / n_{test}) \quad (10)$$

It eliminates dependence on test size and allows comparisons between datasets with different  $n$  values. For example, stratified Dry Bean with LR has  $\log_{10} BF = 1344$  and  $n_{tests} \approx 2.722$ , yielding  $BF d = 0.494$  (moderate evidence per sample). Stratified Iris has  $\log_{10} BF = 12.77$  and  $n_{tests} = 30$ , yielding  $BF d = 0.426$ : comparable evidence per sample. This normalization is analogous to using Pearson's correlation instead of covariance.

#### 6.4. $\Delta H$ as an Indicator of Sufficiency

The information gain  $\Delta H$  complements the BF.  $\Delta H$  values  $\approx 0$  with a low BF indicate a test that is too small to update the prior.  $\Delta H$  values  $\ll 0$  with a very negative BF indicate a dominant prior: the prior is so concentrated that the test, even if perfectly representative, cannot displace it. In this case, the diagnosis points to the model, not the test.

#### 6.5. Practical Application: BF as a Production Monitoring Metric (MLOps)

The model-test compatibility diagnosis translates directly to MLOps pipelines. Once the model is deployed in production, the prior BetaBinomial ( $\alpha 0, \beta 0$ ) is fixed from the training and becomes the reference for expected behavior. Each batch of predictions in production of size  $n_{batches}$  with  $k$  batches of correct predictions is then a new observation that the BF evaluates against that reference.

The proposed monitoring protocol consists of three steps:

**M1. Calculate BF over the batch** With  $(k_{batch}, n_{batch})$  and the fixed prior of the training, we obtain  $\log_{10} BF$  and its normalized version  $BF_d = \log_{10} BF / n_{batch}$ . Normalization is essential in production because batches have variable sizes;  $BF_d$  allows direct comparison of batches of different sizes and the construction of compatibility time series.

**M2. Verify batch sufficiency ( $\Delta H$ ):** If  $\Delta H \approx 0$ , the batch is too small to update the prior and the BF is not informative. This step avoids False alarms from batches that are too small - a frequent problem in systems with low or very irregular traffic.

**M3. Issue alert for baseline deviation** Baseline  $BF_d$  is defined as the average of the  $BF_d$  in the first  $W$  batches post-deployment (calibration period).

An alert is triggered when:

$$BF_{dt} - BF_d BASE > Z\alpha \cdot \sigma BF_{base}, \quad (11)$$

where  $z\alpha$  controls the false positive rate and  $\sigma$  is the standard deviation observed during the calibration period.

A key advantage over existing MLOps tools is that model monitoring platforms like Evidently AI, WhyLabs, and Seldon Alibi Detect calculate shift metrics (PSI, KS, MMD) on test features. This requires the monitoring system to have access to the input space  $X$ , which isn't always possible: in black-box prediction APIs, in models with proprietary, non-exportable features, or when the API contract only exposes the final prediction.

The BF operates exclusively on  $(k_{test}, n_{test})$ , that is, on the model's predictions: it does not require features. This makes it applicable in any scenario where the model is already deployed, without requiring changes to the data pipeline architecture. Additionally, Experiment D demonstrates that the BF detects class bias with an AUC of 0.95, a scenario where the MMD fails completely (AUC of 0.58) even with access to the features.

**Practical alert thresholds.** To implement step M3 in production, the following indicative thresholds are proposed based on the Jeffreys scale adapted to  $BF_d$ :

- **Minor alert** ( $z_\alpha = 1,5 \sigma$ ): activate manual review of the data pipeline. Suggests a minor distributional shift or outlier.
- **Moderate alert** ( $z_\alpha = 2,5 \sigma$ ): pause automatic high-consequence predictions and request retraining.
- **Critical alert** ( $BF_{dt} < 0, i.e., \log_{10} BF < 0$ ): The observed batch is more probable under the random classifier than under the trained model. Immediate action required.

$\sigma$  values are estimated during the post-deployment calibration period ( $W$  first batches).  $W \geq 10$  batches is recommended for robust  $\sigma$  estimation.

## 6.6. Limitations and Future Work

1. **Based diagnostics:** The global Beta-Binomial model collapses all information into  $(k, n)$ . Experiment E implements the class-based extension: for each class  $c$ , an independent prior Beta  $(\alpha 0, c, \beta 0, c)$  models the recall rate of that class in training. The results show that this extension localizes incompatibility: in Dry Bean with shift, BF C4 = -36.3 while BF C2 = +987.9. The remaining limitation is the independence between classes: the model does not capture correlations between the difficulty of similar classes.
2. **Artificial shift vs. real shift:** Feature-ordered splits are proxies for real shift. Validation on datasets with natural domain shifts (temporal, geographic, or data source splits) is the next priority. This work constitutes a proof of concept on controlled datasets; extension to benchmarks such as WILDS [14] or MIMIC-IV temporal splits is future work.
3. **Complex models and deep learning:** The experiments use Logistic Regression and Random Forest. The Beta-Binomial framework is directly applied to any classifier that produces  $(k, n)$  on the test set, including neural networks. However, overconfident models (e.g., transformers without temperature calibration) can systematically produce  $\text{acc train} \approx 1$ , exacerbating the dominant prior problem. The  $\lambda$  factor (Exp F) partially mitigates this effect; pre-computation temperature calibration of the prior is a natural extension.
4. **Structured confusions between classes:** The Beta-Binomial model collapses all the information into global accuracy  $(k, n)$ , losing the structure of the confusion matrix. In scenarios where the shift affects only confusions between similar classes (e.g., classes 3 and 4 in Dry Bean), global BF may not detect it, while BF per class (Exp E) does. A full Dirichlet-Multinomial extension on the confusion matrix, not just on recall per class, would capture these correlations.
5. **Assumption IID:** The Beta-Binomial formulation assumes interchangeability of examples within each batch. It is not directly valid for time series with autocorrelation or for data with spatial or network structure. In those cases, a sequential formulation (Bayesian Kalman filter on  $\theta$ ) would be more appropriate.

## 7. Conclusion

Unlike feature-based shift detectors (KS, MMD), the BF operates solely on the predictive performance of the model, without requiring access to the test features. Seven statistically significant controlled experiments (Wilcoxon,  $N = 20$ ) demonstrate that:

1. The BF, measured as absolute deviation from the reference BF, detects inconsistencies with  $\text{AUC}=0.91\text{--}0.95$  (LR/RF), competitive with prediction-based baselines such as log-loss ( $\text{AUC}=0.953\text{--}0.957$ ) and BBSD ( $\text{AUC}=0.887\text{--}0.992$ ), but using only  $(k, n)$  without `predict_proba`.
2. Class bias is the scenario where the advantage of BF over MMD is most pronounced: MMD is insensitive to class bias, while BF detects it with high sensitivity.
3. The BF falls monotonously with label contamination and crosses the indifference threshold between 25 and 50% noise, depending on the size of the test.
4. In Credit Card Fraud with real-time shift (Exp G), BF falls by 73% (from +1077 to +289) while log-loss and BBSD barely vary, confirming that the BF offers the clearest drift signal with minimal information.

$\Delta H$  metric complements the diagnosis as an indicator of test size sufficiency: values close to zero indicate that the test is too small to update the prior.

The proposed three-step protocol—calculating BF, verifying  $\Delta H$ , and comparing BFd against the baseline—constitutes a production monitoring tool (MLOps) that operates without access to test features. This property makes it applicable to black-box APIs, models with proprietary features, and pipelines where the service contract only exposes the final prediction—scenarios where existing tools (Evidently, WhyLabs, Seldon Alibi) are inapplicable because they require  $\text{test } X$ .

**Future work:** automatic calibration of the  $\lambda$  factor using internal hold-out; extension to naturalistic distributional shift benchmarks such as WILDS [14] or medical domain time datasets (MIMIC-IV); and implementation as an open-source module compatible with MLflow and Kubeflow for native integration into MLOps pipelines.

## Declaration on Generative AI

The authors did not use GenAI during the preparation of this work. The authors reviewed and edited the content as necessary and assume full responsibility for the content of the publication.

## References

- [1] Óscar A. Méndez, J. Camargo, H. Florez, Machine learning operations applied to development and model provisioning, in: *Communications in Computer and Information Science*, Springer, 2024, pp. 73–88.
- [2] H. E. Robbins, An empirical bayes approach to statistics, in: *Breakthroughs in Statistics: Foundations and basic theory*, Springer, 1992, pp. 388–394.
- [3] R. R. Bouckaert, Choosing between two learning algorithms based on calibrated tests, in: *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, 2003, pp. 51–58.
- [4] Z. Dienes, Using bayes to get the most out of non-significant results, *Frontiers in psychology* 5 (2014) 781.
- [5] T. S. I. M. LEARNING, Dataset shift in machine learning, 2009.
- [6] A. Benavoli, G. Corani, J. Demšar, M. Zaffalon, Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis, *Journal of Machine Learning Research* 18 (2017) 1–36.
- [7] B. Efron, C. Morris, Stein’s estimation rule and its competitors—an empirical bayes approach, *Journal of the American Statistical Association* 68 (1973) 117–130.
- [8] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, D. Dennison, Hidden technical debt in machine learning systems, *Advances in neural information processing systems* 28 (2015).
- [9] Z. Lipton, Y.-X. Wang, A. Smola, Detecting and correcting for label shift with black box predictors, in: *International conference on machine learning*, PMLR, 2018, pp. 3122–3130.
- [10] F. Velosa, H. Florez, Edge solution with machine learning and open data to interpret signs for people with visual disability, in: *CEUR Workshop Proceedings*, 2020, pp. 15–26.
- [11] H. Jeffreys, *The theory of probability*, OUP Oxford, 1998.
- [12] E.-J. Wagenmakers, M. Marsman, T. Jamil, A. Ly, J. Verhagen, J. Love, R. Selker, Q. F. Gronau, M. Šmíra, S. Epskamp, et al., Bayesian inference for psychology. part i: Theoretical advantages and practical ramifications, *Psychonomic bulletin & review* 25 (2018) 35–57.
- [13] A. Dal Pozzolo, O. Caelen, R. A. Johnson, G. Bontempi, Calibrating probability with undersampling for unbalanced classification., in: *SSCI*, 2015, pp. 159–166.
- [14] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, et al., Wilds: A benchmark of in-the-wild distribution shifts, in: *International conference on machine learning*, PMLR, 2021, pp. 5637–5664.
- [15] J. Gama, P. Medas, G. Castillo, P. Rodrigues, Learning with drift detection, in: *Brazilian symposium on artificial intelligence*, Springer, 2004, pp. 286–295.
- [16] F. J. Massey Jr, The kolmogorov-smirnov test for goodness of fit, *Journal of the American statistical Association* 46 (1951) 68–78.
- [17] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test, *The journal of machine learning research* 13 (2012) 723–773.