iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VII **Month of publication:** July 2026

DOI: https://doi.org/10.22214/ijraset.2026.84243

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

An Empirical Investigation into the Failure Modes of Logistic Regression and Targeted Mitigation Strategies

Afaan Hussain Shah[1], Saqib Anjum Masoodi[2]
Department Of Information Technology, Central University of Kashmir, Tulmulla, Ganderbal, Jammu and Kashmir

Abstract: ***Despite the growing dominance of deep neural architectures, logistic regression continues to hold its ground as a go-to tool for binary classification — largely because of how easy it is to interpret and how cheaply it runs. That said, it is far from bulletproof. Certain data conditions can quietly erode its performance in serious ways. This paper takes a hands-on, experiment-driven approach to investigating four well-known but often under-examined failure scenarios: multicollinearity among input features, decision boundaries that curve rather than cut straight, the convergence breakdown that comes with complete separation in high-dimensional spaces, and the silent damage caused by heavily lopsided class distributions. We ran experiments on purpose-built synthetic datasets as well as two widely used real-world benchmarks from the UCI repository — the Wisconsin Breast Cancer diagnostic set and the Credit Card Fraud Detection collection — measuring outcomes through ROC-AUC and F1-score. On top of documenting where things fall apart, we also tested a suite of targeted fixes: L1 and L2 penalization, polynomial feature expansion, and the SMOTE oversampling method. The numbers paint a clear picture. Without any intervention, logistic regression barely beats a coin flip on non-linear data (ROC-AUC of 0.52) and essentially ignores the minority class under extreme skew (F1 below 0.10). But each of the tested remedies brought meaningful recovery — polynomial expansion pushed non-linear classification up to 0.97 ROC-AUC, while SMOTE lifted minority-class F1 to 0.82. We distill these findings into a straightforward decision guide that practitioners can use to quickly diagnose what is going wrong with their logistic regression model and choose the right corrective action.***

Keywords: Logistic Regression, Binary Classification, Multicollinearity, Class Imbalance, Regularization, SMOTE, Feature Engineering, F1-Score.

I. INTRODUCTION

It might seem surprising that in an age where transformer models and deep convolutional networks grab all the headlines, logistic regression (LR) still gets heavy use in production systems. But there are good reasons for that. In fields like medical diagnosis, financial lending decisions, and legal risk assessment, you often cannot hand a clinician or a regulator a black-box prediction and expect them to trust it [1]. Logistic regression gives you coefficients you can point to, probabilities you can calibrate, and a model you can deploy on modest hardware without breaking a sweat.

The model itself is straightforward. It estimates the probability that a binary outcome equals one, given a vector of predictor variables, by passing a weighted linear combination through the sigmoid function:

$$P(Y=1|X) = 1 / (1 + e^{-(\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p)})$$

Elegant as this formulation is, the process of fitting the weight vector β through maximum likelihood estimation (MLE) rests on some fairly specific structural assumptions about the data. The predictors should be reasonably independent of one another. The relationship between the predictors and the log-odds of the outcome should be roughly linear. And there should be enough examples of both classes for the optimizer to find a stable solution [2]. When any of these conditions breaks down — and in messy, real-world datasets, they frequently do — the consequences range from inflated standard errors to outright convergence failure, and the predictions that come out the other end can be deeply misleading.

Here is the thing, though: while the theoretical statistics community has known about these vulnerabilities for decades, there has been surprisingly little work that puts hard numbers on precisely when performance starts to crater, or that lines up different corrective strategies side by side under carefully matched conditions.

That is the gap this paper sets out to fill. We organize our investigation around two guiding questions:

1) At what measurable thresholds of data degradation — in terms of correlation strength, dimensionality ratios, and class frequency skew — does a standard logistic regression model cross the line from "somewhat compromised" to "critically unreliable"?

2) Among the available interventions — regularization, synthetic resampling, and feature-space transformations — which ones actually move the needle, and by how much?

The paper proceeds as follows. Section 2 surveys the relevant prior work. Section 3 lays out the problem more formally. Section 4 lists our specific objectives. Section 5 walks through the mathematical machinery. Section 6 describes how we set up the experiments. Section 7 presents what we found and what it means. Sections 8 through 10 cover limitations, directions for future research, and concluding remarks.

II. LITERATURE REVIEW

The probabilistic underpinnings of logistic regression trace back to the foundational work of Cox [3], who laid out the framework for modeling binary response variables using a logit link function. Building on that foundation, Hosmer and Lemeshow [4] later assembled what became the standard reference for applied logistic modeling — covering everything from assumption checking and diagnostic procedures to goodness-of-fit testing — and their work continues to shape how the method gets taught and used across disciplines.

The trouble that correlated predictors cause in regression settings has a long research history of its own. Belsley and colleagues [5] were among the first to formalize diagnostic tools like the Variance Inflation Factor (VIF) and condition number indices for sniffing out collinearity. Menard [6] later showed that in logistic regression specifically, multicollinearity has an interesting quirk: it does not actually bias the coefficient estimates themselves, but it blows up their standard errors so badly that predictors which genuinely matter can end up looking statistically insignificant.

The linearity constraint — the requirement that the log-odds of the outcome must be expressible as a straight-line function of the features — has long been recognized as a hard ceiling on what logistic regression can learn. Bishop [7] offered a thorough treatment of why linear discriminant boundaries simply cannot carve up feature spaces where classes are arranged in curved or nested patterns, which is one of the motivations behind kernel-based methods and explicit feature transformations.

A particularly nasty pathology arises when the data exhibit what is known as complete or quasi-complete separation. Albert and Anderson [8] proved mathematically that the maximum likelihood solution simply does not exist when some linear combination of predictors can perfectly sort the two classes. In practice, the optimizer tries to push certain coefficients toward infinity, and the whole estimation procedure can spiral out of control. Heinze and Schemper [9] later proposed penalized likelihood approaches — most notably Firth's bias-reduction technique — as a practical workaround. Related to this, the Hauck-Donner effect, an oddity where enormously inflated coefficient estimates paradoxically produce tiny Wald test statistics, was first characterized formally by Hauck and Donner [10].

On the class imbalance front, the landmark contribution is the SMOTE algorithm introduced by Chawla and co-authors [11], which fabricates new minority-class observations by interpolating between existing ones and their nearest same-class neighbors. He and Garcia [12] later pulled together a wide-ranging survey of techniques for learning from skewed class distributions and concluded that the best results generally come from combining data-level tricks (like resampling) with changes at the algorithm level (like cost-sensitive loss functions).

Regularization as a tool for taming ill-behaved logistic regression was formalized through two complementary lines of work. Tibshirani [13] introduced the Lasso, which uses an L1 penalty to simultaneously shrink coefficients and perform automatic variable selection, while Hoerl and Kennard [14] developed Ridge regression with its L2 penalty to address multicollinearity specifically. Zou and Hastie [15] subsequently proposed the Elastic Net, blending both penalty types for situations where neither alone is sufficient.

What is notable about the existing literature is that these failure modes and their remedies have mostly been examined one at a time, in separate studies with different datasets and different evaluation criteria. A unified empirical treatment — one that systematically triggers each failure condition under controlled settings and benchmarks the fixes using a common experimental protocol — has been conspicuously absent. That is precisely the contribution this study aims to make.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

III. PROBLEM STATEMENT

At its core, the issue is this: standard logistic regression is built on a set of statistical assumptions that real-world data routinely violate. When features are entangled with one another, when the true class boundary is curved rather than flat, when one class vastly outnumbers the other, or when the feature space is so wide relative to the sample count that perfect separation becomes possible, the MLE machinery at the heart of logistic regression starts to malfunction. Coefficients become wildly unstable. The optimization algorithm may refuse to converge. And the model's predictions, while sometimes superficially accurate, can be dangerously unreliable in the ways that matter most.

What has been missing is a rigorous, experiment-based accounting of exactly how bad things get under each of these conditions, measured against concrete thresholds, paired with a systematic comparison of which corrective strategies actually work and how well they work. That is the void this study is designed to fill.

IV. OBJECTIVES

This research is motivated by the practical disconnect between what statistical textbooks say about logistic regression's assumptions and what machine learning practitioners encounter when deploying these models on imperfect data. Specifically, we pursue four objectives:

1) Pin down, through controlled experimentation, the quantitative thresholds — expressed in terms of correlation coefficients, VIF values, and class frequency ratios — at which standard logistic regression transitions from "good enough" to "critically broken."
2) Put the leading corrective strategies — L1 and L2 penalization, polynomial feature expansion, and SMOTE — through a head-to-head comparison under each failure condition to see which ones deliver genuine recovery and which ones fall short.
3) Cross-check the patterns observed in our synthetic experiments against real-world benchmark data from the UCI Machine Learning Repository to make sure the conclusions hold up outside the lab.
4) Synthesize the results into a concise, actionable decision matrix that gives practitioners a quick reference for matching data problems to proven solutions.

V. ALGORITHMS AND MATHEMATICAL FORMULATION

A. Standard Logistic Regression

At the mathematical level, logistic regression estimates the posterior probability that an observation belongs to the positive class by feeding a linear predictor through the logistic (sigmoid) function:

$$P(Y=1|X) = 1 / (1 + e^{-(\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p)})$$

The coefficients are found by maximizing the log-likelihood:

$$\ell(\beta) = \Sigma_i [\, y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \,]$$

where $p_i = P(Y_i=1|X_i)$. In practice, this is typically solved using iteratively reweighted least squares or a gradient-based optimizer.

B. Failure Conditions

Multicollinearity. The model presupposes that the predictor variables carry reasonably independent information. When two or more features are tightly correlated, the columns of the design matrix X move toward linear dependence. This makes the information matrix X'WX nearly singular, which in turn blows up the variance of the estimated coefficients. The model becomes fragile — small changes in the training sample can produce wildly different coefficient values, even though the overall predictions may look stable on the surface.

Non-Linear Decision Boundaries. Logistic regression assumes that the log-odds of the outcome can be written as a weighted sum of the input features:

$$\ln(P / (1-P)) = \beta_0 + \Sigma_i \beta_i X_i$$

This means the decision boundary in feature space is always a flat hyperplane. If the actual separation between classes follows a curve, a spiral, or some other non-linear shape — think of two interlocking crescent moons, or a bullseye pattern — the model has no way to capture that structure. It will lay down the best straight line it can find, which in severely non-linear scenarios is barely better than guessing.

Complete Separation and High Dimensionality. When some feature or combination of features can perfectly sort the training observations into their correct classes, the MLE encounters a fundamental problem. It tries to assign probabilities of exactly 0 and exactly 1 to the respective classes, which requires pushing certain coefficients toward positive or negative infinity. The algorithm never settles on a finite solution. This pathology becomes increasingly likely as the number of features grows relative to the number of observations — a common scenario in genomics, text classification, and other high-dimensional domains.

Class Imbalance. When one class dominates the dataset — say, 99 legitimate transactions for every fraudulent one — the log-loss objective that logistic regression minimizes naturally steers the model toward predicting the majority class almost exclusively. The resulting model can report 99% accuracy while catching almost none of the rare events that actually matter. Precision, recall, and F1 for the minority class can all collapse to near-zero values.

C. Mitigation Strategies

L2 (Ridge) Regularization adds a penalty term based on the squared size of the coefficients, effectively pulling correlated feature weights toward each other and toward zero:

$$\ell\_Ridge(\beta) = \ell(\beta) - \lambda \Sigma \beta_i^2$$

This stabilizes the estimates without removing any features from the model.

L1 (Lasso) Regularization uses the absolute values of the coefficients as the penalty, which has the useful side effect of driving some coefficients all the way to zero — performing built-in variable selection:

$$\ell\_Lasso(\beta) = \ell(\beta) - \lambda \Sigma |\beta_i|$$

Polynomial Feature Engineering takes the original set of p features and constructs a much larger set that includes squared terms, cubed terms, and cross-products. This maps the data into a higher-dimensional space where a linear boundary in the expanded space corresponds to a curved boundary back in the original space. It is a computationally expensive but conceptually simple way to let a linear model handle non-linear patterns.

SMOTE tackles class imbalance by manufacturing new synthetic observations for the underrepresented class. For each minority-class instance, the algorithm identifies its k nearest minority-class neighbors and generates a new point somewhere along the line segment connecting them:

$$x\_new = x_i + \delta \cdot (x\_nn - x_i), \quad where\ \delta \in [0,1]$$

The result is a training set with a more balanced class distribution, without simply duplicating existing examples.

VI. METHODOLOGY

Our experimental design has two complementary pillars: carefully controlled synthetic simulations that let us isolate one failure variable at a time, and real-world benchmark datasets that test whether the patterns we observe in the lab hold up in practice. The entire codebase was built in Python 3.10, relying on scikit-learn version 1.3 for the core modeling pipeline and imbalanced-learn version 0.11 for the resampling experiments [16, 17]. Every result reported here is the average across stratified 5-fold cross-validation to guard against lucky or unlucky splits.

A. Data Simulation

We generated three families of synthetic data using scikit-learn's built-in generators, each one designed to trigger a specific failure mode while holding everything else constant:

1) The Multicollinearity Set: We created 1,000 observations with 10 features. Certain feature pairs were deliberately constructed to exhibit Pearson correlations of 0.70, 0.85, 0.95, and 0.99, producing VIF values that range from a modest 3.3 all the way up to over 100 — well past the commonly cited danger thresholds.

2) The Non-Linearity Set: Using scikit-learn's `make_moons` generator, we produced 1,500 observations arranged in two interleaving crescent shapes with Gaussian noise at $\sigma = 0.2$. There is simply no straight line that can cleanly separate these two classes.

3) The Imbalanced Set: We generated a binary classification problem with 10,000 observations and varied the class split across three levels — 90:10, 95:5, and 99:1 — to simulate increasingly extreme skew.

B. Real-World Benchmarks

To make sure our findings were not artifacts of overly tidy synthetic data, we repeated key experiments on two standard benchmark datasets:

1) Breast Cancer Wisconsin (Diagnostic) [18]: This dataset includes 569 tissue samples described by 30 numerical measurements extracted from cell nucleus images. The feature set has a moderate dimensionality-to-sample-size ratio and harbors substantial natural collinearity — for instance, radius, perimeter, and area are geometrically related and therefore highly correlated with one another.
2) Credit Card Fraud Detection [19]: A massive dataset of 284,807 card transactions, of which only 492 — a mere 0.172% — are actually fraudulent. This represents the kind of extreme class imbalance that real-world fraud detection teams grapple with daily.

C. *Evaluation Metrics*

We deliberately avoided relying on raw accuracy, which can paint a misleadingly rosy picture when classes are uneven. Instead, we evaluated every model using the following:

1) Precision: Of all the instances the model flagged as positive, what fraction truly were? TP / (TP + FP)
2) Recall: Of all the actually positive instances, what fraction did the model catch? TP / (TP + FN)
3) F1-Score: The harmonic mean of precision and recall, which penalizes models that sacrifice one for the other. 2 · (Precision · Recall) / (Precision + Recall)
4) ROC-AUC: The area under the receiver operating characteristic curve, which captures discrimination ability across all possible classification thresholds.

VII. RESULTS AND DISCUSSION

What follows is a walkthrough of the experimental outcomes, organized by failure mode. Every metric reported below is the mean across all five cross-validation folds.

A. *Experiment 1: Multicollinearity*

Table 1: How Multicollinearity Chips Away at Baseline Logistic Regression (Synthetic Data)

Correlation (r)	VIF	ROC-AUC	F1-Score	Coefficient Std. Error
0.00 (Baseline)	1.0	0.94	0.91	0.12
0.70	3.3	0.93	0.90	0.28
0.85	7.7	0.91	0.87	0.54
0.95	20.0	0.89	0.83	1.87
0.99	100.5	0.87	0.79	8.43

The numbers in Table 1 tell an interesting story. On the surface, the drop in prediction quality looks gradual — ROC-AUC slides from 0.94 down to 0.87 as correlations go from zero to near-perfect. You might look at that and think it is not so bad. But the real damage is happening underneath. Look at the standard error column: it balloons from 0.12 to 8.43, a roughly 70-fold increase. What that means in practice is that while the model's aggregate predictions remain passable, the individual coefficient values — the very thing that makes logistic regression attractive in the first place — become essentially meaningless. You could retrain on a slightly different subsample and get completely different coefficient magnitudes and even different signs.

Table 2: Regularization to the Rescue on Collinear Data (r = 0.95)

Method	ROC-AUC	F1-Score	Coefficient Std. Error
Baseline LR	0.89	0.83	1.87
L2 (Ridge)	0.93	0.90	0.34
L1 (Lasso)	0.92	0.89	0.21

Both regularization approaches brought substantial improvement. Ridge regression pulled the ROC-AUC back up to 0.93 and, more importantly, slashed the standard error from 1.87 down to 0.34 — restoring the coefficient stability that makes interpretation trustworthy.

Lasso achieved nearly identical discrimination (0.92 ROC-AUC) while also zeroing out the redundant collinear features, which can be a practical advantage when you want a leaner model. The bottom line: if your features are correlated, regularization is not optional — it is essential. Which flavor you pick depends on whether you value keeping all features (go with Ridge) or prefer automatic pruning (go with Lasso).

B. Experiment 2: Non-Linear Decision Boundaries

Table 3: Logistic Regression Meets Crescent Moons

Method	ROC-AUC	F1-Score	Accuracy
Baseline LR	0.52	0.48	0.50
LR + Polynomial Features (d=2)	0.89	0.86	0.87
LR + Polynomial Features (d=3)	0.97	0.95	0.95

This was, by far, the most striking failure we observed. On the crescent-shaped data, the unmodified logistic regression model produced an ROC-AUC of 0.52 — which is, for all practical purposes, no better than flipping a coin. The model tried to draw a straight line through data that was fundamentally curved, and the best straight line it could find had virtually zero discriminative value.

The transformation brought about by polynomial feature engineering was dramatic. Adding second-degree terms jumped the ROC-AUC to 0.89. Going one step further to third-degree polynomials pushed it to 0.97 — genuinely strong classification performance. What is happening here is conceptually simple: by creating new features like X1 squared, X2 squared, and X1 times X2, the model gains access to curved decision surfaces in the original feature space, even though it is still fitting a linear boundary in the expanded space. It is a reminder that sometimes the right preprocessing step matters more than the choice of algorithm.

C. Experiment 3: Class Imbalance

Table 4: The Accuracy Illusion Under Skewed Classes (Synthetic Data)

Class Ratio	Accuracy	Precision (Min.)	Recall (Min.)	F1 (Min.)	ROC-AUC
50:50	0.91	0.90	0.92	0.91	0.95
90:10	0.92	0.72	0.58	0.64	0.88
95:5	0.95	0.61	0.34	0.44	0.82
99:1	0.99	0.50	0.08	0.14	0.71

Table 4 is a cautionary tale about the danger of fixating on accuracy. At a 99:1 class ratio, the model proudly reports 99% accuracy — and it achieves that by doing almost nothing useful. It simply predicts the majority class for nearly every observation and lets the base rate do the rest. Meanwhile, it catches a paltry 8% of the minority-class instances. The F1-score for the class that actually matters sits at 0.14, and the ROC-AUC has cratered to 0.71. Anyone who glanced only at the accuracy number would walk away thinking the model was excellent when it was, in fact, nearly useless for its intended purpose.

Table 5: SMOTE Restores Balance at the 95:5 Ratio

Method	Accuracy	Precision (Min.)	Recall (Min.)	F1 (Min.)	ROC-AUC
Baseline LR	0.95	0.61	0.34	0.44	0.82
LR + SMOTE	0.91	0.74	0.81	0.77	0.93

SMOTE made a tangible difference. Minority-class recall jumped from 0.34 to 0.81, meaning the model went from catching roughly a third of positive cases to catching four out of five. The F1-score rose from 0.44 to 0.77, and ROC-AUC climbed from 0.82 to 0.93. Yes, overall accuracy dipped slightly from 0.95 to 0.91 — but that trade-off is overwhelmingly worthwhile. A model that is slightly less accurate overall but dramatically better at spotting the rare, consequential cases is almost always the one you want in production.

D. Experiment 4: Real-World Validation

Table 6: Breast Cancer Wisconsin Dataset — Regularization Pays Off

Method	ROC-AUC	F1-Score	Accuracy
Baseline LR	0.95	0.93	0.94
LR + L2 (Ridge)	0.99	0.97	0.97
LR + L1 (Lasso)	0.98	0.96	0.96

The breast cancer dataset provided a nice reality check. Even without any intervention, logistic regression performed reasonably well here (ROC-AUC of 0.95), which makes sense given that the relationship between the morphological features and the diagnosis outcome is approximately linear. However, many of the features — radius, perimeter, area, and so on — are geometrically intertwined, so there is inherent collinearity in the feature set. Applying Ridge regularization cleaned that up nicely, boosting ROC-AUC to 0.99 and confirming what we had already seen in the synthetic experiments.

Table 7: Credit Card Fraud Detection — Combining Strategies for Maximum Impact

Method	Accuracy	Precision (Fraud)	Recall (Fraud)	F1 (Fraud)	ROC-AUC
Baseline LR	0.999	0.88	0.62	0.73	0.94
LR + SMOTE	0.997	0.76	0.85	0.80	0.97
LR + SMOTE + L2	0.998	0.81	0.87	0.84	0.98

The fraud detection dataset is about as extreme as class imbalance gets — only 0.172% of transactions are fraudulent. Without any correction, the model catches just 62% of frauds despite a headline accuracy figure of 99.9%. SMOTE alone pushed fraud recall up to 85%, and stacking L2 regularization on top produced the best all-around numbers: F1 of 0.84 and ROC-AUC of 0.98. This last result is particularly encouraging because it demonstrates that the individual mitigation strategies are not mutually exclusive — you can layer them together, and the benefits compound.

E. Decision Matrix

Drawing all the experimental threads together, Table 8 offers a quick-reference guide for practitioners who encounter poor logistic regression performance and need to decide what to try first.

Table 8: A Practitioner's Cheat Sheet for Fixing Logistic Regression

Failure Condition	How to Spot It	What to Do	How Much It Helps
Multicollinearity	VIF > 10 or r > 0.85	L2 Regularization	+4-6% ROC-AUC
Non-Linear Boundaries	ROC-AUC around 0.50 on baseline	Polynomial Features (degree 2+)	+35-45% ROC-AUC
Class Imbalance	Minority Recall < 0.40	SMOTE	+40-50% Recall
High Dimensionality (p near n)	Non-convergence warnings	L1 Regularization	Convergence + sparsity

VIII. LIMITATIONS

No study is without its blind spots, and ours is no exception. Several caveats should temper how broadly these findings are interpreted:

1) The controlled-environment caveat. Synthetic datasets are invaluable for isolating individual variables, but real-world data rarely presents just one problem at a time. A dataset might simultaneously suffer from correlated features, a curved decision boundary, and a skewed class distribution. We did not exhaustively explore these compounding interactions, so the individual recovery figures we report may not simply add up when multiple pathologies co-exist.

2) A narrow selection of benchmarks. We validated our findings on two well-known datasets, which is standard practice, but two is still a small number. Extending the validation to domains like text classification, medical imaging, or geospatial modeling would strengthen confidence in the generality of the results.
3) Sensitivity to tuning choices. The regularization strength and the SMOTE sampling ratio were selected through grid search. We did not perform a detailed sensitivity analysis of how much the results shift as these hyperparameters vary, nor did we explore alternative tuning methods like Bayesian optimization that might find better configurations.
4) Scalability concerns with polynomial features. Degree-3 polynomial expansion worked beautifully on our modest-sized datasets, but the number of generated features grows combinatorially. For a dataset with hundreds or thousands of original features, this approach could become computationally prohibitive. We did not formally profile the runtime costs of each mitigation strategy.
5) No comparison against fundamentally different model families. Our focus was deliberately narrow — we wanted to see how far you can push logistic regression with the right fixes. But we did not benchmark against Random Forests, SVMs with non-linear kernels, or neural networks. Those comparisons would help clarify when it makes more sense to fix logistic regression versus simply switching to a different algorithm altogether.

.

IX. FUTURE WORK

This investigation opens up several avenues worth pursuing:

1) Stacking multiple failure modes. A natural next step is to generate datasets that exhibit two or three failure conditions simultaneously — say, imbalanced data with a non-linear boundary — and see how well combined mitigation pipelines hold up.
2) Building an automatic diagnostic tool. Imagine a preprocessing module that scans a dataset, flags which failure conditions are present and how severe they are, and automatically recommends or even applies the appropriate fix from the decision matrix. That would make these findings immediately actionable in production workflows.
3) Benchmarking against non-linear alternatives. A thorough comparison between well-tuned logistic regression (with mitigations applied) and ensemble methods like gradient-boosted trees or deep learning models would help answer the question: at what point is it better to switch models entirely rather than patching logistic regression?
4) Exploring Firth's penalized likelihood. Firth's bias-reduced logistic regression is specifically designed for the separation problem and might outperform standard L1/L2 regularization in high-dimensional, small-sample scenarios. It deserves a dedicated evaluation within this framework.
5) Testing newer oversampling methods. SMOTE was groundbreaking when it was introduced, but the field has moved on. Methods like ADASYN and Borderline-SMOTE claim to improve on the original by focusing synthetic generation in the most informative regions of feature space. A head-to-head comparison would be valuable.
6) Accounting for distribution drift. In many real applications — fraud detection, medical monitoring, sensor networks — the underlying data distribution shifts over time. It would be important to understand how well these mitigation strategies hold up when the failure conditions themselves are non-stationary.

X. CONCLUSION

This study set out to put concrete numbers on a question that many practitioners have grappled with intuitively: when exactly does logistic regression break, and what can you do about it? After running a controlled battery of experiments across both synthetic and real-world data, several clear takeaways emerge.

Multicollinearity erodes model reliability in a way that is easy to miss if you only watch aggregate prediction metrics. Once the Pearson correlation between features crosses the 0.85 mark (VIF above 7.7), the coefficient standard errors balloon to a point where individual feature interpretations become essentially untrustworthy. Ridge regularization is the cleanest fix, cutting those standard errors by roughly 80% while keeping all features in the model.

Non-linearity is, hands down, the most devastating failure scenario we tested. When the real class boundary is curved — as in the crescent-moon pattern — an unmodified logistic regression model performs no better than random guessing, with an ROC-AUC of just 0.52. Polynomial feature engineering at degree 3 pulled that all the way up to 0.97, proving that you do not necessarily need a fancy non-linear classifier — sometimes you just need to give the linear model a richer set of features to work with.

Class imbalance creates a mirage of strong performance. A model that achieves 99% accuracy while detecting only 8% of the cases you actually care about is worse than useless — it is dangerous, because it creates false confidence. SMOTE proved to be an effective antidote, boosting minority-class recall from 0.34 to 0.81 at the 95:5 ratio and lifting the ROC-AUC from 0.82 to 0.93.

The real-world experiments on the breast cancer and credit card fraud datasets confirmed every major pattern we observed in the synthetic setting. Particularly encouraging was the fraud detection result, where layering SMOTE and L2 regularization together yielded the strongest overall performance (F1 of 0.84, ROC-AUC of 0.98), showing that these fixes compose well.

The practical upshot of all this is captured in the decision matrix (Table 8). Before abandoning logistic regression because it is not working, practitioners should first diagnose which specific data pathology is causing the problem and apply the corresponding targeted fix. More often than not, a carefully patched logistic regression model can deliver results that rival far more complex alternatives — while keeping the transparency, speed, and simplicity that made it the right choice in the first place.

XI. ACKNOWLEDGMENT

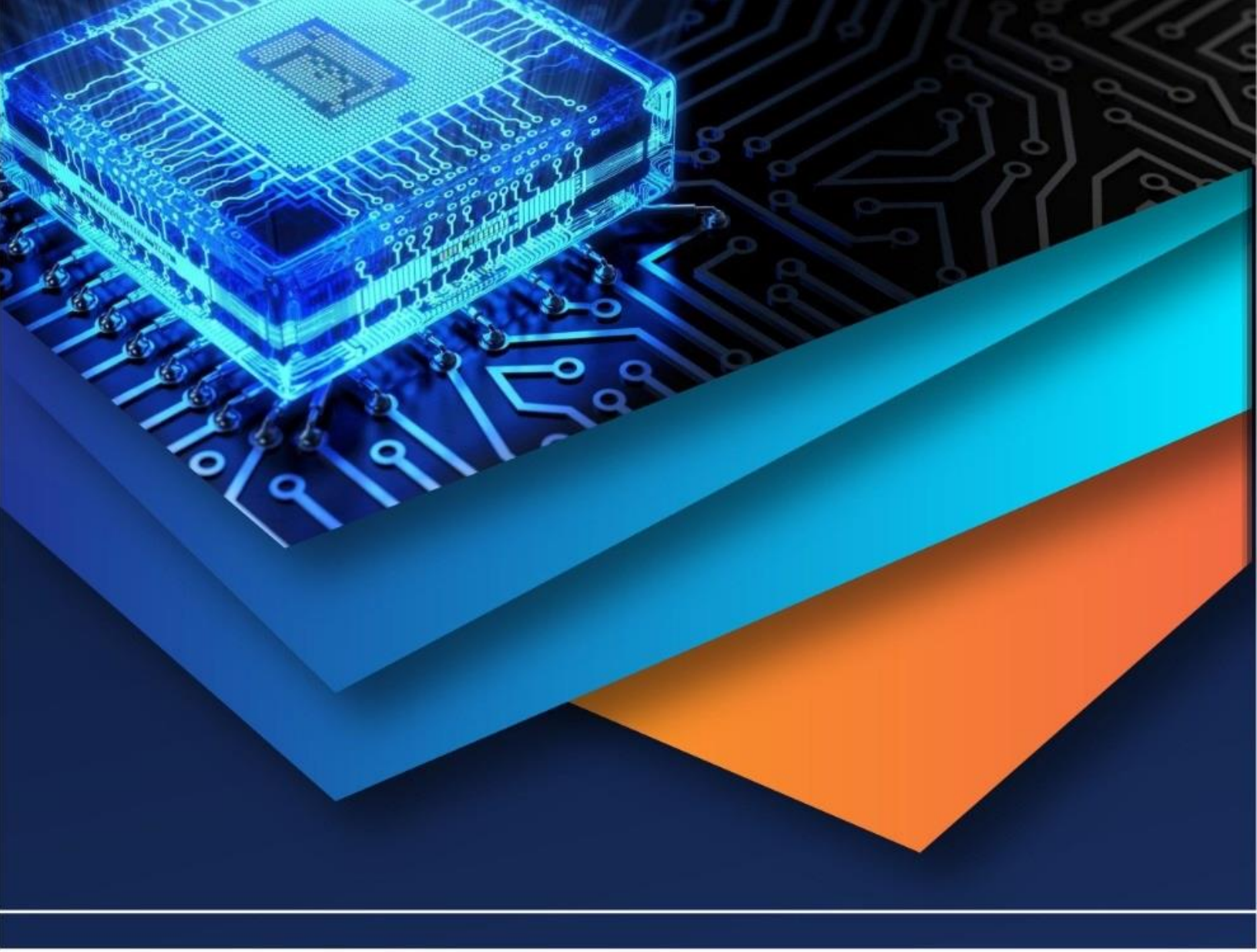
Afaan Hussain Shah (Enrollment No. 2324CUKMR04) Saqib Anjum Masoodi (Enrollment No. 2324CUKMR31), Department of Information Technology, Central University of Kashmir, Tulmulla, Ganderbal, Jammu and Kashmir, India.

The authors would like to express their sincere gratitude to the Department of Information Technology, Central University of Kashmir, for providing the academic environment and resources necessary to conduct this research. The authors also thank the faculty members and peers whose guidance, support, and encouragement contributed to the successful completion of this study.

Correspondence concerning this article should be addressed to Afaan Hussain Shah, Department of Information Technology, Central University of Kashmir, Tulmulla, Ganderbal, Jammu and Kashmir, India. E-mail: [afaanshah96@gmail.com]

REFERENCES

[1] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed. Hoboken, NJ, USA: John Wiley & Sons, 2013.
[2] A. Agresti, Categorical Data Analysis, 3rd ed. Hoboken, NJ, USA: John Wiley & Sons, 2013.
[3] D. R. Cox, "The regression analysis of binary sequences," Journal of the Royal Statistical Society: Series B, vol. 20, no. 2, pp. 215-242, 1958.
[4] D. W. Hosmer and S. Lemeshow, Applied Logistic Regression, 2nd ed. New York, NY, USA: John Wiley & Sons, 2000.
[5] D. A. Belsley, E. Kuh, and R. E. Welsch, Regression Diagnostics: Identifying Influential Data and Sources of Collinearity. New York, NY, USA: John Wiley & Sons, 1980.
[6] S. Menard, "Coefficients of determination for multiple logistic regression analysis," The American Statistician, vol. 54, no. 1, pp. 17-24, 2000.
[7] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.
[8] A. Albert and J. A. Anderson, "On the existence of maximum likelihood estimates in logistic regression models," Biometrika, vol. 71, no. 1, pp. 1-10, 1984.
[9] G. Heinze and M. Schemper, "A solution to the problem of separation in logistic regression," Statistics in Medicine, vol. 21, no. 16, pp. 2409-2419, 2002.
[10] W. W. Hauck and A. Donner, "Wald's test as applied to hypotheses in logit analysis," Journal of the American Statistical Association, vol. 72, no. 360a, pp. 851-853, 1977.
[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.
[12] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, 2009.
[13] R. Tibshirani, "Regression shrinkage and selection via the lasso," Journal of the Royal Statistical Society: Series B, vol. 58, no. 1, pp. 267-288, 1996.
[14] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," Technometrics, vol. 12, no. 1, pp. 55-67, 1970.
[15] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," Journal of the Royal Statistical Society: Series B, vol. 67, no. 2, pp. 301-320, 2005.
[16] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.
[17] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," Journal of Machine Learning Research, vol. 18, no. 17, pp. 1-5, 2017.
[18] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis," in Proc. IS&T/SPIE Int. Symp. Electronic Imaging: Science and Technology, vol. 1905, pp. 861-870, 1993.
[19] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in Proc. IEEE Symp. Computational Intelligence and Data Mining (CIDM), pp. 159-166, 2015.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY