Confidence intervals for the classification accuracy metrics when oversampling the very minor class: a "black" box simulation study

Yury Festa[1] and Henry Penikas*[2]

[1]Independent researcher, Moscow, Russia
[2]Bank of Russia, Research and Forecasting Department, Moscow, Russia.

February 19, 2025

Abstract

AI applications in finance including those for the probability of default modeling largely involve using ML classification tools. Oversampling the very minor (very underrepresented) class of defaulted borrowers seems to be a must-be-done step always. However, by crunching more than a thousand of confidence intervals for the classification accuracy metrics, we demonstrate when such oversampling is worth engaging in. Moreover, we argue to what portion of total initial sample size such oversampling should be carried out. Our findings are valuable primarily for the credit risk modeling and Internal Ratings Based (IRB) banks, but are not limited to those and have general applications for the binary classifications in ML domain.

Key Words: credit risk, precision, recall, F1, classification, clustering, segmentation, IRB.

JEL Codes: C12, C25, C52, C58, D81, G32.

Disclaimer

The views expressed herein are solely those of the authors. The content and results of this research should not be considered or referred to in any publications as the Bank of Russia's official position, official policy, or decisions. Any errors in this document are the responsibility of the authors.

*Corresponding author, penikas@gmail.com.

I prefer true but imperfect knowledge,
even if it leaves much indetermined and unpredictable,
to a pretence of exact knowledge that is likely to be false.

Hayek (1974), Nobel Prize lecture

1 Introduction

The Basel Committee report BCBS (2017) might be named the first formal recognition of the material artificial intelligence (AI) proliferation in the finance domain. Formally, it even led to the introduction of the new terms like FinTech, RegTech, and SupTech. At the time the committee saw only technological risks posed by the proliferation of AI, machine learning (ML), and advanced data analytics considered jointly. As a result, the committee recommended strengthening the information technologies (IT) with which the bank is equipped, see (BCBS, 2017, pp. 28).

Since then the AI/ML use made that significant progress that the associated risks stopped being limited solely by IT ones. More conceptual issues arose. Those include the ethical ones whether an algorithm should be allowed or not to discriminate one cohort of customers to the detriment (rarely - to the benefit) of another. This led to the discussion of the ethical probability of default (PD) models in papers like Fuster et al. (2018); Szepannek and Luebke (2021). The European Parliament extended the discussion by making an unprecedented step and publishing a pan-European AI regulation act, see Europarliament (2023).

So far, it seems that methodologically everything is clear with the development of AI in finance, and it is only the issue of the available (sufficient) computational capacities. Such thoughts gave rise to the terms of *GPU-rich* and *GPU-poor* companies distinguishing companies which have enough access to the needed graphical processing unit (GPU) capacities and those which do not have, see The Economist (2024).

However, today seems to be right the time when we may fall into the fundamental trap created by our obsession with the exact prediction and hence recommendation skills of AI modules driven by the underlying ML solutions. The nature of the trap is as follows. The recent ML trend allows software to elaborate own programming codes and models, in particular (though still far from ideally targeted ones as developed by experienced coders). The AI solution of interest is likely to continue reprogramming the specific model as far as its output performance (accuracy) metrics outpaces that of the previous one. Such a process goes on as in most cases it is the point estimates of the performance metrics which rise, though sometimes at a tiny growth rate. From the outside perspective such an improvement process in addition vastly consumes GPU power making any company GPU-poor in essence.

Nevertheless, the improvement process is not as endless as it seems and as it was in the legend when Achilles failed to outrun the turtle. In fact, most models become similar when the model performance metrics reach a particular threshold for a combination of classes and features. Such similarity is well captured by the confidence intervals (CI) for the performance metrics, which unfortunately are not that wide-spread though well-known in the probability theory. Hence, if the AI algorithm for a credit scoring or fraud detection in finance reached the stage when the upper boundary of the accuracy metrics CI is almost equal to one (to 100%), it is clear that any novel model cannot discriminate poor borrowers from good ones any better (unless there happens a region-wide shock and overall model prediction quality deteriorates). This could mean that AI software may get rise in efficiency by not crunching the code and numbers any longer and by economizing the GPU capacity for other tasks.

The use of confidence intervals for the performance metrics of ML models in finance is not novel. Moreover, the cases when one of the two classes is materially underrepresented is also known (consider the term *low-default portfolio (LDP)*, for instance). Oversampling minor class is

a typical industry solution. However, no one, to the best of our knowledge, studied the confidence intervals evolution for the performance metrics of the models in finance when such oversampling is undertaken. We intend to close this gap.

As a preview of our findings, we show that excessive oversampling (at the extreme when equalizing the proportions of the minor and major classes) leads to the rise in the width of the confidence intervals of the performance metrics making models more indistinguishable from each other, and by overall sacrifizing the model performance quality. The practical implication from here is to oversample at a limited degree. Then and only then the model developer (or AI software supposedly at the near future) may be able to evidence the true improvement in the model performance.

To explain how we arrive at our findings, we start with the literature review in Section 2. We describe the methodology in Section 3. The findings follow in Section 4. We conclude in Section 5.

2 Literature review

AI applications in finance, though numerous, can be broadly grouped into several groups of which classification tasks continue occupying important place. Those tasks might include distinguishing good and bad borrowers, clients prone to churn and not, online users willing to choose a product or not, fraudsters and general users. Solving classification (properly discriminating) in-between these two groups forms the basis for further recommendation system development.

Hence, it is vitally important to be efficient in solving classification tasks when applying AI and ML in finance. Seems lots has been discussed about it in Mirkin (2016); Raschka and Mirjalili (2019), for instance. However, gaps still exist. Those relate to situations when one of two classes is materially underrepresented (such a class might be called a *very minor* one, while the residual class is a major one). A fast, but not always worthy typical solution is to oversample. This is why we intend to study consequences of such a step given often omitted specifics for the confidence intervals when applied to the classification accuracy metrics.

To do so, we first discuss the papers when dealing with minor classes is not a one-off case. Namely, it is the domain of probability of default (PD) modeling and developing PD models for banks specifically. Nevertheless, the findings are of value to other areas, including inter alia (cyber-)fraud detection. Second, we remind approaches to handling minor class when it might be assumed to be underrepresented in a non-systematic manner. This is where the suggestion to oversample the minor class is being born. Third, we focus on how to choose the best classification model as it is exactly the criteria which are intended to be improved when oversampling. Forth, we rehearse the importance of monitoring the confidence intervals for the classification metrics, not limited to their mean values.

2.1 PD modeling

The first formal probability of default (PD) models were proposed in the papers by Beaver (1966); Altman (1968); Ohlson (1980). Authors of these papers used quite countable number of observations driven by the computational capabilities of the first computers. Often those equaled couple of dozens company-year (or just company) observations. Moreover, the samples of defaulted and non-defaulted companies typically equalled in size giving no rise to the issue of handing a minor class.

Since then software and financial services industries evolved that much that PD models started being considered as part of the financial regulation. Formally, the Basel Committee on Banking Supervision (BCBS) allowed them as a part of the Basel II Internal Ratings-Based (IRB) approach, see BCBS (2006). Prior to formal adoption, the committee published a comprehensive survey of progress in classification models development, and more specifically to that of PD models in BCBS

(2000). Highly likely it was that the PD model conceptual approval by the international financial regulation standards setter of BCBS triggered the research boom in the area.

As a result, we come across the use of conventional econometric and multivariate statistical analysis tools to develop PD models as discussed by Kumar and Ravi (2007); Altman (2018). Same time the use of ML tools gains its popularity as can be seen from the following non-exhausting list of papers: Chen et al. (2006); Fantazzini and Figini (2009); Korol and Korodi (2010); Tinoco and Wilson (2014); Geng et al. (2015); Jabeur and Fahmi (2018); Shibitov and Mamedli (2019); Qu et al. (2019); Dendramis et al. (2020); Kim et al. (2020); Moscatelli et al. (2020); Kim et al. (2021); Faraj et al. (2021); Pang et al. (2021); Merćep et al. (2021); Liu et al. (2022).

PD models were developed for many localities. To name a few, Jabeur and Fahmi (2018) considered France, Chen et al. (2006); Liu et al. (2022) - China, Altman et al. (2008) - the UK, Tian and Yu (2017) - Japan, Bisogno et al. (2018) - the EU, Kristóf and Virág (2020) - Hungary, Merćep et al. (2021) - Croatia.

Most academic papers present PD models for the retail borrowers as the segment is typically characterized by the enormous number of observations and defaults. PD models for corporate borrowers appear less often, while banks are the rarest research objects. For instance, they are handled in the following relevant works: Bräuning et al. (2020); Durand et al. (2021) for the EU, Yuksel et al. (2015) for Turkey, Shrivastava et al. (2020) for India, Kočenda and Iwasaki (2022) for Japan, Kocagil et al. (2002); Moody's Analytics (2016); Cole et al. (2020) for the USA, Obeid (2021) for the Persian Gulf countries, and Cheong and Ramasamy (2019); Kristóf (2021) for others. Relevant reviews are available at Kumar and Ravi (2007); Citterio (2020).

Table 1: Why dealing with a tiny class is important?

#	Paper	Class	Country	Method	Freq.	Pred.Hor.	Period	# X vars	# obs. (N)	# Def. (D)	DR = D / N
1	Kocagil et al. (2002)	Banks	USA	probit	Y	1Y, 5Y	1982-2002	15 –> 6	140000	400	0,0029
2	Moody's Analytics (2016)	Banks	World (90x)	[probit]	Y	1Y, 5Y	1988-2012	6	33000	200	0,0061
3	Shibitov and Mamedli (2019)	Banks	Russia	ML	M	1-9M	2014-18	35 –> 721	34096	354	0,0104
4	Ferriani et al. (2019)	Banks	Italy	logit	Q	4-6Q	2008-16	18	9571	195	0,0204

Note: Freq. - data frequency.

The reason for such rarity of the PD model for banks can be vividly seen from the illustrative table 1. Nowadays, as well as 20 years ago, financial institutions (FI) tend mostly not to default. The proportion of defaulted cases at maximum approaches 2% of the total sample, being as small as less than half of the percentage point (see last column of table 1). This is why financiers tend to call the FI segment a *low default portfolio (LDP)*. AI/ML practitioners eagerly see the problem (defaulted) cases in the segment as the *very minor* class with the non-defaulters being a very major one.

2.2 Missing data and oversampling

Though the FI segment is not rich in defaults, the financiers solicited PD models for the segment. There are several solutions on how to act, according to (Raschka and Mirjalili, 2019, pp. 267-270):

- to oversample the minor class, Liu (2021); Nunes et al. (2021);
- to undersample the major class.
- to imput missings, Audigier et al. (2021);

Koziarski (2021) opts for a combination of over- and undersampling. However, oversampling is grounded on the strong assumptions. According to Rubin (1976) classification, it is assumed that the data (default cases) is missing either completely at random (MCAR), or just at random (MAR). However, Carreras et al. (2021) argues that if the data is of MCAR type, then oversampling is not needed, as one is to add pure noise not-impacting the model of interest.

On the contrary, the possibility of data being missing not at random (MNAR) is rarely checked. To be fair, in the absence of extra defaults, the feasibility of such verification by itself is under question. Pereira et al. (2019) offers arguments to ignore MNAR, as it stems from situations when the data was not collected or was wrongly collected via a survey. Financial default data has a more regular nature, and only extreme force-major events might trigger systematic unaccounting of many default cases. Alternatively, when wishing to handle MNAR cases, one may drift towards Heymans and Twisk (2022) who suggests modeling the missing data. But to do so, one should properly study such MNAR cases, then calibrate the data generating process parameters. One can do the latter step only by using the available limited (LDP) cases. Hence, we also neglect the possibility of MNAR observations here.

2.3 Classification accuracy (model performance) metrics

ML practitioners tend to oversample minor class as rule of thumb. Our objective here is to demonstrate cases when such oversampling is worth undertaking and when it is not. To answer this question, we should first inquire what objective is targeted when oversampling. The ML practitioners seek to improve (increase) the model quality (its performance metrics), i.e., the model developers wish the model to better discriminate (segment, classify, cluster) the incoming data into two classes (in case of PD model into defaulters and non-defaulters). The industry-standard is to look at precision, recall, accuracy, F1 indicators. The respective formulas are available in eq. (1) - (4).

$$Precision = \frac{TP}{TP + FP}, \tag{1}$$

where TP, FP are illustrated in Table 2.

$$Recall = \frac{TP}{TP + FN}, \tag{2}$$

where FN is presented in Table 2.

$$Accuracy = \frac{AUROC\ for\ the\ developed\ model}{AUROC\ for\ the\ perfect\ model}, \tag{3}$$

where the numerator and denominator are computed after deducting the common surface (triangle) under the bisector line. We may recommend Engelmann et al. (2003) as one of the earlier papers in the finance domain for more details on AUROC use for the PD modeling.

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = \frac{2}{(1/Recall) + (1/Precision)}. \tag{4}$$

Table 2: Stylized default (success) prediction matrix to analyze model accuracy.

	Actual		
Predicted	S (D)	F (ND)	Total
S (D)	**True** positives (TP)	False positives (FP)	· P
F (ND)	False negatives (FN)	**True** negatives (TN)	· N
Total			n

Note (conventional suggested by us for the purposes of the current study): minor class : S - success, or D - default; major class: F - failure, or ND - non-default; n stands for the total number of observations.

2.4 Confidence intervals for proportions

We have evidenced above that PD models are well-studied, accuracy metrics are also commonly known. However, the problem - inter alia with the growing number of papers published and offering the better discriminating PD models - is that authors get obsessed with the improvement solely based on the mean values (point estimates) of the classification metrics of interest, e.g., (Faraj et al., 2021, p. 24, Tab. 2), (Kim et al., 2021, p. 170, Tab. 4), (Pang et al., 2021, p. 10), (Merćep et al., 2021, p. 10, Table 1 - p. 12, Table 6), (Song et al., 2021, p.1489, Table 1), (Liu et al., 2022, p. 10, Tab. 8).

Nevertheless, we should not forget that the performance metrics combine the number of realisations of a random variable, (often a dummy flag taking one in case of default and zero otherwise). They differ from each other in a way of such combination. Disregarding the mode of combination, the accuracy metrics by construction are still random variables in themselves. It means that the mere dominance (excess in arithmetic terms) of one point estimate over another may correspond to a probabilistically equal values. To correctly judge upon the superiority of a particular model, when comparing PD models, one has to look at the confidence intervals of performance metrics, not limited to their point estimates. Moreover, as every accuracy metrics is a proportion by construction ranging from zero to one, one should specifically look at the confidence intervals for (binomial) proportions.

The development of the confidence intervals (CIs) for proportions has passed through the following stages:

1. Wald CI, or normal approximation, see formula (5);

$$CI^N = (S/n) + / - \gamma_{\alpha/2} \cdot \sqrt{\frac{(S/n) \cdot [1 - (S/n)]}{n}}, \tag{5}$$

   where $\gamma_{\alpha/2} = N^{-1}(\alpha/2)$ is the quantile of the Normal (Gaussian) distribution at the $\alpha/2$ significance level, n is the total number of observations, S is the number of successes (F is the number of failures, so that $n = S + F$).

2. Wilson CI, see formulas (6)

$$CI^W = \frac{S + (\gamma^2_{\alpha/2}/2)}{n + \gamma^2_{\alpha/2}} + / - \gamma_{\alpha/2} \cdot \frac{\sqrt{[(SF)/n^2] + (\gamma^2_{\alpha/2}/4)}}{n + \gamma^2_{\alpha/2}}. \tag{6}$$

   where $\gamma_{\alpha/2} = \lambda = 2$ is recommended in most cases, see (Wilson, 1927, p. 212).

3. Clopper-Pearson (beta) CI, see (Dunnigan, 2008, p. 3), formulas (7), (8);

$$CI^{CP}_L = \frac{1}{1 + \frac{n-S+1}{S} \cdot F_{2(n-S+1),2S,\alpha/2}}, \tag{7}$$

   where $F_{u,v,\gamma}$ is the F-distribution with (u, v) degrees of freedom valued at γ significance level.

$$CI^{CP}_U = \frac{\frac{S+1}{n-S} F_{2(S+1),2(n-S),\alpha/2}}{1 + \frac{S+1}{n-S} F_{2(S+1),2(n-S),\alpha/2}}. \tag{8}$$

   Orawo (2021) notes that Clopper-Pearson CI is more conservative, but wider than it is sufficient.

4. Agresti-Coull (AC) CI from (Agresti and Coull, 1998, p. 120), see formula (9);

$$CI^{AC} = \frac{S + (\gamma^2_{\alpha/2}/2)}{n + \gamma^2_{\alpha/2}} + / - \gamma_{\alpha/2} \cdot \frac{\sqrt{(SF) \cdot [1 + (\gamma^2_{\alpha/2}/2)] + (\gamma^4_{\alpha/2}/4)}}{n + \gamma^2_{\alpha/2}}. \tag{9}$$

5. Jeffreys CI, see formulas (10), (11);

$$CI^J_L = Beta(\alpha/2; S + 1/2, n - S + 1/2), \tag{10}$$

$$CI^J_U = Beta(1 - \alpha/2; S + 1/2, n - S + 1/2), \tag{11}$$

where $Beta(\alpha, a_1, a_2)$ is the α-quantile of the Beta distribution with parameters a_1 and a_2, see (Brown et al., 2001, p. 108, eq. (7), (8)); L and U indicate lower and upper boundaries of the confidence interval.

Brown et al. (2001) above all recommend using Jeffreys interval instead of normal approximation, as well as instead of Wilson's and Agresti-Coull's ones.

3 Simulation experiment design

3.1 Concept

We wish to study how confidence intervals for the classification accuracy metrics evolve under various scenarios. We look at three starting values of the minor class (e.g., default rates, DR): 0.1, 3.0, 10.0% of the total number of observations. These portions are the starting (baseline) values. We oversample them to reach up to 50% of the initial number of observations. For instance, take a $DR = 0.001$ (0.1%). The total number of observations is 20k, it yields us with 20 default cases and 19 080 non-default ones. When oversampling to 50% of the initial set, we get 10k defaults instead of just 20 ones. Hence, the new sample size is 10k + 19080 = 39080 observations. As for the DR=0.1% we run extra oversampling iterations to 10, 20, 30, 40% to be able to identify the threshold at which the CI width starts changing.

We use ten core features (independent factors) to delineate minor class observations from the major ones. We consider four possible factor combinations. We start with the availability of all 10 core features, then we add extra redundant 5 features to have 15 in total. Next, we deduct 5 core ones from the initial set and proceed with 5 core features. Last, we add 5 redundant features to the 5 core ones left from the previous stage.

For each model we evaluate four classification metrics as presented in subsection 2.3. For each of the metrics we present five confidence intervals (CIs) discussed in subsection 2.4. Hence, we derive five CI widths as differences between the CI lower boundary (_L) and its upper one (_U).

When the CI width augments, the models become less distinguishable. Hence, it becomes more difficult to offer another model statistically (probabilitistically) outpacing the value of the current accuracy metrics value. Thus, we are interested in cases when the CI width shrinks. Then models are more divisible. Having built a new model it is more likely to evidence that it is superior to the existing one *all else being equal*.

3.2 Parameter specification

We use the **make_classification** package in Python to generate initial data with the default flags (zeros and ones) and accompanying values of the so called (hypothetical) *informative* risk-drivers (core features). The raw features' values are drawn from the standard normal distribution. A cut threshold is applied to a linear combination of factors in order to obtain the targeted proportion of the minor class.

We add noise to our classification via a $flip\_y$ parameter. It is the portion of observations to which the class (default flag) is assigned randomly. By default, its value is 0.01. We took it equal to 0.5.

To oversample, we use an **imblearn.over_sampling** library with the *SMOTE* method, Chawla et al. (2002). We change the *sampling_strategy* parameter to obtain new portions of the minor (resampled) class. The new observations are not mere duplicates of the existing ones. They have the features values drawn from the empirical (non-parametric) distribution fitted for the minor class observations.

To build a model, we use **GridSearchCV** package in Python. We maximize F1 metrics and report confidence intervals for it. Overall, we fit 64 models and look at 1.2k confidence intervals.

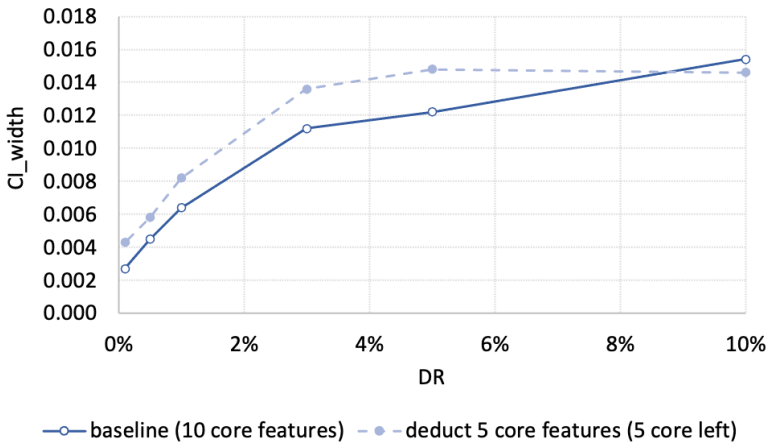
The data simulation details are available in Annex A.

4 Findings

Here we enlist the key findings which we obtain from our simulation experiment (table 3 contains the details on the average widths of the five considered CIs for the F1 metrics):

1. The width of the CI is proportionate to the share of the minor class, e.g., the lower the default rate (DR) is, the narrower the CI is, compare D1 to D7 (1.5% vs 0.3%) in table 3; see also figure 1.

Figure 1: Higher portion of minor class imply wider CI.



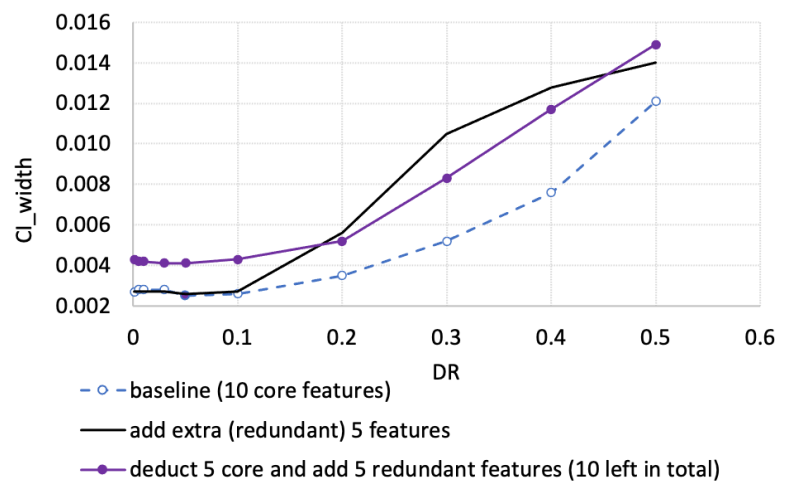
2. When the portion of the minor class (DR) is low (below 5%), making some core features unavailable leads to the increase in the CI width, compare D3 to F3 (1.1% vs 1.3%) and D7 to F7 (0.3% vs 0.4%) in table 3. However, when the portion is larger (e.g., 10%), we may observe reduction of the CI width, compare D1 to F1 (1.54% vs 1.46%) in table 3.

3. Adding more noise (extra redundant features) widens the CI when oversampling from a very tiny class to equal proportions case (from 0.1% to 50%), compare D16 to E16 (1.2% vs 1.4%) and F16 to G16 (1.1% vs 1.5%) in table 3. In other cases, we do not trace neither material deterioration, nor improvement in CI width.

4. Oversampling to equal class shares (50:50%) mostly often leads to deterioration (CI widening), compare D3 to D6 (1.1% vs 1.3%) and rows 1 to 2 in table 3. However, in a realistic set-up (column G) when we know part of core drivers and also include several redundant ones, oversampling not a very minor class ($DR \approx 3\%$) might improve the situation and make CI narrower, compare F3 to F6 (1.36% vs 1.28%) and G3 to G6 (1.4% vs 1.1%) in table 3.

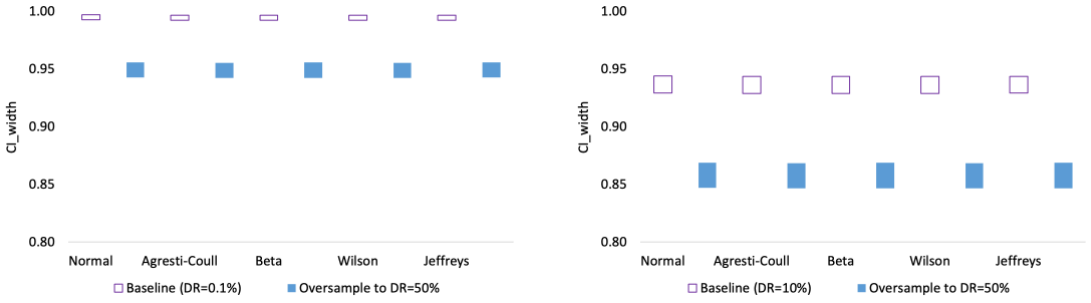
5. Oversampling the very minor class might be reasonable when considering moderate pace of resampled observations. For instance, oversampling DR of 0.1% enables to slightly reduce the CI width when the portion reaches 3-5%, but above that the CI width starts rising, compare rows 7 to 10 and 11 in table 3; see also figure 2.

Figure 2: Oversampling very minor class improves (narrows) CI, but for mild resampling.



6. Oversampling often leads not merely to CI widening, but also to overall model performance deterioration. As a result, CI shifts down, see figure 3.

Figure 3: Oversampling very minor to equal portions not only widens the CI, but also drastically reduces the mean performance (shifts the CI down).



However, we do not notice material differences in the application of different CI types, all of the five ones move in tandem for both low and high initial portions of the minor class, see figure 3.

5 Conclusion and practical implications

AI is nowadays thought of being an indispensable element of future progress in finance. Such progress encapsulates the proliferation of the ML models' use for the numerous classification tasks, including the discrimination of good from bad borrowers, i.e., for the development of the probability of default (PD) models.

We show that PD model developers often face a challenge when coming across an underrepresented (minor) class. As a remedy, they solicit industry-wide practice of oversampling the minor class. This is why we focus on PD models, though our findings spread far beyond PD modeling, and are generally applicable to any binary classification task.

We manage to dig deeper into the properties of models when the underlying data is oversampled. Importantly, we show the thresholds to which it is worth oversampling the minor class given its initial portion. For instance, when the portion is moderately small one (around 3% of the total sample size), one may benefit from oversampling it to 50%. However, when the initial class is very tiny (around 0.1% of total number of observations), it might be worth oversampling only to 3-5% of the total number of entries. Moreover, we argue that such a gain in (narrowing of) confidence interval for the PD model performance might be achieved with the trade-off by losing the overall model performance (the CI mid value materially goes down).

We offered a statistical table which might be used by practitioners as a guide when to oversample the data or not. The enclosed programming code in Python allows gathering the equivalent answer for any combination of initial portion of the minor class, number of core and redundant features, the considered oversampling proportions.

Annex

Table 3: Summary of simulation experiments.

A	B	C	D	E	F	G
row #	Model Type	DR	baseline (10 core features)	add extra (redundant) 5 features	deduct 5 core features (5 core left)	deduct 5 core and add 5 redundant features (10 left in total)
1	Baseline	0.1	0.0154	0.0154	0.0146	0.0146
2	Oversampled	0.5	0.01636	0.01642	0.02184	0.02188
3	Baseline	0.03	0.0112	0.0112	0.0136	0.0136
4	Oversampled	0.05	0.0115	0.0115	0.0138	0.0138
5	Oversampled	0.1	0.0117	0.0117	0.0143	0.0143
6	Oversampled	0.5	0.0125	0.0125	0.0128	0.011
7	Baseline	0.001	0.0027	0.0027	0.0043	0.0043
8	Oversampled	0.005	0.0028	0.0027	0.0042	0.0042
9	Oversampled	0.01	0.0028	0.0027	0.0042	0.0042
10	Oversampled	0.03	0.0028	0.0027	0.0041	0.0041
11	Oversampled	0.05	0.0025	0.0026	0.0041	0.0041
12	Oversampled	0.1	0.0026	0.0027	0.0042	0.0043
13	Oversampled	0.2	0.0035	0.0056	0.0052	0.0052
14	Oversampled	0.3	0.0052	0.0105	0.0052	0.0083
15	Oversampled	0.4	0.0076	0.0128	0.0052	0.0117
16	Oversampled	0.5	0.0121	0.014	0.011	0.0149

Note: DR - default rate (proportion, share of the minor class). Underlying confidence interval boundaries are presented in the Technical Annex (available from the authors upon request).

References

Agresti, A. and Coull, B. A. (1998). Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*, 52:119–126. http://dx.doi.org/10.1080/00031305.1998.10480550, restricted access; https://math.unm.edu/~james/Agresti1998.pdf, open access, accessed on Dec. 30, 2023.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, 23(4):589–609.

Altman, E. I. (2018). A fifty-year retrospective on credit risk models, the Altman Z-score family of models and their applications to financial markets and managerial strategies. *Journal of Credit Risk*, 14(4):1–34. http://doi.org/10.21314/JCR.2018.243, restricted access; https://mebfaber.com/wp-content/uploads/2020/11/Altman_Z_score_models_final.pdf, open access, accessed on Dec. 26, 2023.

Altman, E. I., Sabato, G., and Wilson, N. (2008). The value of qualitative information in sme risk management. https://pages.stern.nyu.edu/~ealtman/SME_EA_GS_NW.pdf.

Audigier, V., Niang, N., and Resche-Rigon, M. (2021). Clustering with missing data: which imputation model for which cluster analysis method? https://arxiv.org/pdf/2106.04424.pdf. Online; accessed 15 January 2022.

BCBS (2000). Credit ratings and complementary sources of credit quality information. Working Paper No. 3; https://www.bis.org/publ/bcbs_wp3.pdf, open access, accessed on June 11, 2024.

BCBS (2006). Basel II: International Convergence of Capital Measurement and Capital Standards: A Revised Framework - Comprehensive Version. https://www.bis.org/publ/bcbs128.pdf, open access, accessed on June 11, 2024.

BCBS (2017). Implications of fintech developments for banks and bank supervisors. Basel Committee for Banking Supervision Consultative Paper, URL: https://www.bis.org/bcbs/publ/d415.htm, free access, accessed on Aug. 13, 2022.

Beaver, W. H. (1966). Financial ratios as predictors of failure. *Journal of accounting research*, pages 71–111.

Bisogno, M., Restaino, M., and Di Carlo, A. (2018). Forecasting and preventing bankruptcy: A conceptual review. *African journal of business management*, 12(9):231–242.

Bräuning, M., Malikkidou, D., Scalone, S., and Scricco, G. (2020). A new approach to early warning systems for small European banks. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 551–562. Springer.

Brown, L. D., Cai, T. T., and DasGupta, A. (2001). Interval estimation for a binomial proportion. *Statistical Science*, 16:101–133. https://www.jstor.org/stable/2676784, restricted access; http://www-stat.wharton.upenn.edu/~lbrown/Papers/2001a%20Interval%20estimation%20for%20a%20binomial%20proportion%20(with%20T.%20T.%20Cai%20and%20A.%20DasGupta).pdf, open access, accessed on Dec. 26, 2023.

Carreras, G., Miccinesi, G., Wilcock, A., Preston, N., Nieboer, D., Deliens, L., Groenvold, M., Lunder, U., van der Heide, A., Baccini, M., van der Heide, A., Korfage, I. J., Rietjens, J. A. C., Jabbarian, L. J., Polinder, S., van Delden, H., Kars, M., Zwakman, M., Deliens, L., Verkissen,

M. N., Eecloo, K., Faes, K., Pollock, K., Seymour, J., Caswell, G., Wilcock, A., Bramley, L., Payne, S., Preston, N., Dunleavy, L., Sowerby, E., Miccinesi, G., Bulli, F., Ingravallo, F., Carreras, G., Toccafondi, A., Gorini, G., Lunder, U., Červ, B., Simonič, A., Mimić, A., Kodba-Čeh, H., Ozbič, P., Groenvold, M., Arnfeldt, C., Thit Johnsen, A., and ACTION consortium (2021). Missing not at random in end of life care studies: multiple imputation and sensitivity analysis on data from the ACTION study. *BMC Medical Research Methodology*, 21(1):13. https://doi.org/10.1186/s12874-020-01180-y.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:1–26. https://doi.org/10.1613/jair.953, open access, accessed on Aug. 27, 2024.

Chen, J., Marshall, B. R., Zhang, J., and Ganesh, S. (2006). Financial distress prediction in China. *Review of Pacific Basin Financial Markets and Policies*, 09(02):317–336. https://doi.org/10.1142/S0219091506000744, restricted access.

Cheong, C. W. and Ramasamy, S. (2019). Bank failure: A new approach to prediction and supervision. *Asian Journal of Finance & Accounting*, 11:111–140.

Citterio, A. (2020). Bank failures: review and comparison of prediction models. https://ssrn.com/abstract=3719997.

Cole, R. A., Taylor, J., and Wu, Q. (2020). Predicting bank failures using a simple dynamic hazard model. *Available at SSRN 1460526*.

Dendramis, Y., Tzavalis, E., and Cheimarioti, A. (2020). Measuring the Default Risk of Small Business Loans: Improved Credit Risk Prediction using Deep Learning. Athens University of Economics and Business, School of Economic Sciences Working Paper No. 12-2020; https://www.dept.aueb.gr/sites/default/files/allWP-12-20-Dendram-Tzaval-Cheimar-12-11-20_0.pdf, open access, accessed on May 31, 2024.

Dunnigan, K. (2008). Confidence interval calculation for binomial proportions. https://www.mwsug.org/proceedings/2008/pharma/MWSUG-2008-P08.pdf, open access, accessed on May 29, 2024.

Durand, P., Le Quang, G., et al. (2021). What do bankrupcty prediction models tell us about banking regulation? Evidence from statistical and learning approaches. https://xtra.economix.fr/pdf/dt/2021/WP_EcoX_2021-2.pdf?1.0.

Engelmann, B., Hayden, E., and Tasche, D. (2003). Testing rating accuracy. *Risk*, pages 82–86. https://www.researchgate.net/publication/215991100_Testing_Rating_Accuracy, open access, accessed on Dec. 26, 2023.

Europarliament (2023). EU AI act: first regulation on artificial intelligence. The use of artificial intelligence in the EU will be regulated by the AI act, the world's first comprehensive AI law. find out how it will protect you. https://www.europarl.europa.eu/news/en/headlines/society/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence, open access, accessed on May 29, 2024.

Fantazzini, D. and Figini, S. (2009). Random survival forests models for SME credit risk measurement. *Methodology and Computing in Applied Probability*, 17:29–45. https://doi.org/10.1007/s11009-008-9078-2, restricted access.

Faraj, A. A., Mahmud, D. A., and Rashid, B. N. (2021). Comparison of different ensemble methods in credit card default prediction. *UHD Journal of Science and Technology*, 5:20–25. https://doi.org/10.21928/uhdjst.v5n2y2021.pp20-25, open access, accessed on Dec. 31, 2023.

Ferriani, F., Cornacchia, W., Farroni, P., Ferrara, E., Guarino, F., and Pisanti, F. (2019). An early warning system for less significant Italian banks. https://www.bancaditalia.it/pubblicazioni/qef/2019-0480/QEF_480_19.pdf. Bank of Italy Occasional Paper No. 480.

Fuster, A., Goldsmith-Pinkham, P., Ramadorai, T., and Walther, A. (2018). Predictably unequal? The effects of machine learning on credit markets. https://doi.org/10.1111/jofi.13090, open access, accessed on Dec. 30, 2023.

Geng, R., Bose, I., and Chen, X. (2015). Prediction of financial distress: An empirical study of listed chinese companies using data mining. *European Journal of Operational Research*, 241(1):236–247. https://doi.org/10.1016/j.ejor.2014.08.016, restricted access.

Hayek, F. A. v. (1974). The pretence of knowledge. Nobel Lecture; https://www.nobelprize.org/prizes/economic-sciences/1974/hayek/lecture/, free access, accessed on Aug. 05, 2022.

Heymans, M. W. and Twisk, J. W. R. (2022). Handling missing data in clinical research. *Journal of Clinical Epidemiology*, 151:185–188. https://doi.org/10.1016/j.jclinepi.2022.08.016, open access, accessed on Jan. 23, 2024.

Jabeur, S. B. and Fahmi, Y. (2018). Forecasting financial distress for French firms: a comparative study. *Empirical Economics*, 54:1173–1186. https://doi.org/10.1007/s00181-017-1246-1,restricted access.

Kim, H., Cho, H., and Ryu, D. (2020). Corporate default predictions using machine learning: Literature review. *Sustainability*, 12:1–11. https://doi.org/10.3390/su12166325, open access, accessed on Dec. 31, 2023.

Kim, H., Cho, H., and Ryu, D. (2021). Predicting corporate defaults using machine learning with geometric-lag variables. *Investment Analyst Journal*, 50:161–175. https://doi.org/10.1080/10293523.2021.1941554, open access, accessed on Dec. 31, 2023.

Kocagil, A., Reyngold, A., Stein, R., and Ibarra, E. (2002). Moody's RiskCalc™ Model for Privately-Held U.S. Banks. http://www.rogermstein.com/wp-content/uploads/riskcalc-usbanks.pdf.

Kočenda, E. and Iwasaki, I. (2022). Bank survival around the world: A meta-analytic review. *Journal of Economic Surveys*, 36:108–156.

Korol, T. and Korodi, A. (2010). Predicting bankruptcy with the use of macroeconomic variables. *Economic Computation and Economic Cybernetics Studies and Research*, 44:201–219. https://www.researchgate.net/publication/289639976_Predicting_bankruptcy_with_the_use_of_macroeconomic_variables, limited access.

Koziarski, M. (2021). Potential anchoring for imbalanced data classification. *Pattern Recognition*, 120:108114. https://doi.org/10.1016/j.patcog.2021.108114.

Kristóf, T. (2021). Bank failure prediction in the COVID-19 environment. *Asian Journal of Economics and Finance*, 3(1):157–171.

Kristóf, T. and Virág, M. (2020). A comprehensive review of corporate bankruptcy prediction in Hungary. *Journal of Risk and Financial Management*, 13(2):35.

Kumar, P. R. and Ravi, V. (2007). Bankruptcy prediction in banks and firms via statistical and intelligent techniques–a review. *European journal of operational research*, 180(1):1–28. https://doi.org/10.1016/j.ejor.2006.08.043, restricted access.

Liu, J. (2021). A minority oversampling approach for fault detection with heterogeneous imbalanced data. *Expert Systems With Applications*, 184:115492. https://doi.org/10.1016/j.eswa.2021.115492, restricted access.

Liu, Y., Yang, M., Wang, Y., Li, Y., Xiong, T., and Li, A. (2022). Applying machine learning algorithms to predict default probability in the online credit market: Evidence from China. *International Review of Financial Analysis*, 79:101971. https://doi.org/10.1016/j.irfa.2021.101971, restricted access.

Merćep, A., Mrčela, L., Birov, M., and Kostanjčar, Z. (2021). Deep neural networks for behavioral credit rating. *Entropy*, 23:806–816. https://dx.doi.org/10.3390/e23010027, open access, accessed on May 29, 2024.

Mirkin, B. (2016). *Clustering: A Data Recovery Approach*. Chapman & Hall, 2nd edition. https://doi.org/10.1201/9781420034912, open access, accessed on Jan. 10, 2024.

Moody's Analytics (2016). RiskCalcTM Banks 4.0. https://www.moodysanalytics.com/-/media/products/riskcalc-banks-4.pdf. The publication year is not explicitly disclosed, though we may refer to the copyright year.

Moscatelli, M., Parlapiano, F., Narizzano, S., and Viggiano, G. (2020). Corporate default forecasting with machine learning. *Expert Systems with Applications*, 161:113567.

Nunes, A. R., Morais, H., and Sardinha, A. (2021). Use of learning mechanisms to improve the condition monitoring of wind turbine generators: A review. *Energies*, 14:7129. https://doi.org/10.3390/en14217129, restricted access.

Obeid, R. (2021). Bank failure prediction in the arab region using logistic regression model. https://www.amf.org.ae/sites/default/files/publications/2021-12/bank-failure-prediction-arab-region-using-logistic-regression-model.pdf. Arab Monetary Fund (Working Paper No. 7-2021), Available online.

Ohlson, J. A. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*, pages 109–131.

Orawo, L. A. (2021). Confidence intervals for the binomial proportion: A comparison of four methods. *Open Journal of Statistics*, 11:806–816. https://doi.org/10.4236/ojs.2021.115047, open access, accessed on May 29, 2024.

Pang, S., Hou, X., and Xia, L. (2021). Borrowers' credit quality scoring model and applications, with default discriminant analysis based on the extreme learning machine. *Technological Forecasting and Social Change*, 165:120462. https://doi.org/10.1016/j.techfore.2020.120462, restricted access.

Pereira, R. C., Santos, M., Rodrigues, P., and Henriques Abreu, P. (2019). MNAR Imputation with Distributed Healthcare Data. In *Progress in Artificial Intelligence*, volume 11805, pages 184–195. https://doi.org/10.1007/978-3-030-30244-3_16.

Qu, Y., Quan, P., Lei, M., and Shi, Y. (2019). Review of bankruptcy prediction using machine learning and deep learning techniques. *Procedia Computer Science*, 162:895–899. https://doi.org/10.1016/j.procs.2019.12.065, open access, accessed on Dec. 30, 2023.

Raschka, S. and Mirjalili, V. (2019). *Python Machine Learning. Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*. Packt, 3rd edition. https://www.amazon.com/Python-Machine-Learning-scikit-learn-TensorFlow/dp/1789955750, restricted access; codes: https://github.com/rasbt/python-machine-learning-book-3rd-edition, open access, accessed on Dec. 30, 2023.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592. https://doi.org/10.1093/biomet/63.3.581.

Shibitov, D. and Mamedli, M. (2019). The finer points of model comparison in machine learning: forecasting based on Russian banks' data. http://www.cbr.ru/content/document/file/87572/wp43_e.pdf. Online; accessed on September 08, 2020.

Shrivastava, S., Jeyanthi, P. M., and Singh, S. (2020). Failure prediction of indian banks using smote, lasso regression, bagging and boosting. *Cogent Economics & Finance*, 8(1):1729569.

Song, C., Wu, J., Zhu, L., and Deng, H.-p. (2021). Research on an adaptive upsampling algorithm for photovoltaic panel segmentation. *Journal of Chinese Computer Science*, pages 1485–1491. http://xwxt.sict.ac.cn/EN/Y2021/V42/I7/1485, open access, accessed on Dec. 26, 2023.

Szepannek, G. and Luebke, K. (2021). Facing the challenges of developing fair risk scoring models. *Frontiers in Artificial Intelligence*. https://doi.org/10.3389/frai.2021.681915, open access, accessed on Dec. 30, 2023.

The Economist (2024). Can Nvidia be dethroned? Meet the startups vying for its crown. a new generation of AI chips is on the way. https://www.economist.com/business/2024/05/19/can-nvidia-be-dethroned-meet-the-startups-vying-for-its-crown, open access, accessed on May 29, 2024.

Tian, S. and Yu, Y. (2017). Financial ratios and bankruptcy predictions: An international evidence. *International Review of Economics & Finance*, 51:510–526.

Tinoco, M. and Wilson, N. (2014). Financial distress and bankruptcy prediction among listed companies using accounting, market and macroeconomic variables. *International Review of Financial Analysis*, 30:394–419. https://doi.org/10.1016/j.irfa.2013.02.013, restricted access; https://core.ac.uk/download/pdf/20482286.pdf, open access, accessed on Dec. 30, 2023.

Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212. http://www.jstor.org/stable/2276774, open access, accessed on Dec. 30, 2023.

Yuksel, S., Dincer, H., and Hacioglu, U. (2015). Camels-based determinants for the credit rating of turkish deposit banks. *International Journal of Finance & Banking Studies*, 4(4):1–17.

TECHNICAL ANNEX TO THE PAPER
(to be provided upon request)

Confidence intervals for the classification accuracy metrics when oversampling the very minor class: a "black" box simulation study

Yury Festa[1] and Henry Penikas[12]

[1]Independent researcher, Moscow, Russia
[2]Bank of Russia, Research and Forecasting Department, Moscow, Russia.

Version dated **February 19, 2025**

[1]Corresponding author, penikas@gmail.com.

A Synthetic data simulation in Python

```
import os
import sys

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression, SGDClassifier, LinearRegression
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis, QuadraticDiscri
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.datasets import make_classification
from sklearn.metrics import classification_report
from sklearn import metrics

from statsmodels.stats.proportion import proportion_confint
from sklearn.model_selection import GridSearchCV

import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

import statistics
from math import sqrt
from imblearn.datasets import make_imbalance

import warnings
warnings.filterwarnings('ignore')


model_list = ['normal', 'agresti_coull', 'beta', 'wilson', 'jeffreys']

def explode_confusion_matrix(cm):

    TP = cm[0][0]
    FP = cm[0][1]
    FN = cm[1][0]
    TN = cm[1][1]

    return TP, FP, FN, TN

def return_metrics(TP, FP, FN, TN):

    acuuracy = (TP+TN)/(TP+TN+FP+FN)
    precision = TP/(TP+FP)
    recall = TP/(TP+FN)
    f1 = 2*((precision*recall)/(precision + recall))

    return acuuracy, precision, recall, f1
```

```
def count_success(TP, FP, FN, TN):

    success_accuracy = TP + TN
    unsuccess_accuracy = FP + FN

    success_precision = TP
    unsuccess_precision = FP

    success_recall = TP
    unsuccess_recall = FN

    success_f1 = 2*TP
    unsuccess_f1 = FN+FP

    return success_accuracy, unsuccess_accuracy, success_precision, unsuccess_precisio

def make_bounds(success_accuracy, unsuccess_accuracy, success_precision, unsuccess_pre

    model_list_ = []

    proportion_accuracy_ci_low_ = []
    proportion_precision_ci_low_ = []
    proportion_recall_ci_low_ = []
    proportion_f1_ci_low_ = []

    proportion_accuracy_ci_up_ = []
    proportion_precision_ci_up_ = []
    proportion_recall_ci_up_ = []
    proportion_f1_ci_up_ = []

    for model in model_list:

        proportion_accuracy_ci_low, proportion_accuracy_ci_up = proportion_confint(suc
        proportion_precision_ci_low, proportion_precision_ci_up = proportion_confint(s
        proportion_recall_ci_low, proportion_recall_ci_up = proportion_confint(success
        proportion_f1_ci_low, proportion_f1_ci_up = proportion_confint(success_f1, suc

        model_list_.append(model)

        proportion_accuracy_ci_low_.append(proportion_accuracy_ci_low)
        proportion_precision_ci_low_.append(proportion_precision_ci_low)
        proportion_recall_ci_low_.append(proportion_recall_ci_low)
        proportion_f1_ci_low_.append(proportion_f1_ci_low)

        proportion_accuracy_ci_up_.append(proportion_accuracy_ci_up)
        proportion_precision_ci_up_.append(proportion_precision_ci_up)
        proportion_recall_ci_up_.append(proportion_recall_ci_up)
        proportion_f1_ci_up_.append(proportion_f1_ci_up)
```

```
    models_frame = pd.DataFrame()

    models_frame['approximation'] = model_list_

    models_frame['accuracy_ci_low'] = proportion_accuracy_ci_low_
    models_frame['accuracy_ci_up'] = proportion_accuracy_ci_up_

    models_frame['precision_ci_low'] = proportion_precision_ci_low_
    models_frame['precision_ci_up'] = proportion_precision_ci_up_

    models_frame['recall_ci_low'] = proportion_recall_ci_low_
    models_frame['recall_ci_up'] = proportion_recall_ci_up_

    models_frame['f1_ci_low'] = proportion_f1_ci_low_
    models_frame['f1_ci_up'] = proportion_f1_ci_up_

    models_frame['accuracy_width'] = models_frame['accuracy_ci_up'] - models_frame['ac
    models_frame['precision_width'] = models_frame['precision_ci_up'] - models_frame[
    models_frame['recall_width'] = models_frame['recall_ci_up'] - models_frame['recall
    models_frame['f1_width'] = models_frame['f1_ci_up'] - models_frame['f1_ci_low']

    return models_frame

def make_classification_matrix(X, y):

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_st

    # GridSearchCV params

    param_grid_logit = [
        {
            'penalty' : ['l1', 'l2', 'elasticnet', 'none'],
            'C' : np.logspace(-4, 4, 20),
            'solver' : ['lbfgs','newton-cg','liblinear','sag','saga'],
            'max_iter' : [10, 100, 500, 1000]

        }
    ]

    param_grid_sgdc = [
        {
            'loss' : ['hinge', 'log', 'modified_huber', 'squared_hinge', 'perceptron']
            'penalty' : ['l1', 'l2', 'elasticnet'],
            'alpha' : [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000],
            'learning_rate' : ['constant', 'optimal', 'invscaling', 'adaptive'],
            'class_weight' : [{1:0.5, 0:0.5}, {1:0.4, 0:0.6}, {1:0.6, 0:0.4}, {1:0.7,
            'eta0' : [1, 10, 100]
        }
    ]
```

```
param_grid_lda = [
    {
        'solver': ['svd', 'lsqr', 'eigen']
    }
]

param_grid_qda = [
    {
        'reg_param': [0.1, 0.2, 0.3, 0.4, 0.5]
    }
]


param_grid_rfc = [
    {
        'n_estimators': [25, 50, 100, 150],
        'max_features': ['sqrt', 'log2', None],
        'max_depth': [3, 6, 9, 12],
        'max_leaf_nodes': [3, 6, 9, 12]
    }
]


param_grid_mlpc = [
    {
        'hidden_layer_sizes': [(10,30,10),(20,)],
        'activation': ['tanh', 'relu'],
        'solver': ['sgd', 'adam'],
        'alpha': [0.0001, 0.05],
        'learning_rate': ['constant','adaptive']
    }
]


clf_logit = GridSearchCV(LogisticRegression(), param_grid = param_grid_logit, cv =
y_pred_logit = clf_logit.predict(X_test)
cnf_matrix_logit = metrics.confusion_matrix(y_test, y_pred_logit)

clf_sgdc = GridSearchCV(SGDClassifier(), param_grid = param_grid_sgdc, cv = 3, ver
y_pred_sgdc = clf_sgdc.predict(X_test)
cnf_matrix_sgdc = metrics.confusion_matrix(y_test, y_pred_sgdc)

clf_lda = GridSearchCV(LinearDiscriminantAnalysis(), param_grid = param_grid_lda,
y_pred_lda = clf_lda.predict(X_test)
cnf_matrix_lda = metrics.confusion_matrix(y_test, y_pred_lda)

clf_qda = GridSearchCV(QuadraticDiscriminantAnalysis(), param_grid = param_grid_qd
y_pred_qda = clf_qda.predict(X_test)
cnf_matrix_qda = metrics.confusion_matrix(y_test, y_pred_qda)
```

```
    clf_rfc = GridSearchCV(RandomForestClassifier(), param_grid = param_grid_rfc, cv =
    y_pred_rfc = clf_rfc.predict(X_test)
    cnf_matrix_rfc = metrics.confusion_matrix(y_test, y_pred_rfc)

    clf_mlpc = GridSearchCV(MLPClassifier(max_iter=100), param_grid = param_grid_mlpc
    y_pred_mlpc = clf_mlpc.predict(X_test)
    cnf_matrix_mlpc = metrics.confusion_matrix(y_test, y_pred_mlpc)

    return cnf_matrix_sgdc, cnf_matrix_logit, cnf_matrix_lda, cnf_matrix_qda, cnf_matr

def make_classification_matrix_over(X_over, y_over, X_src, y_src):

    X_train, X_test = X_over, X_src

    y_train, y_test = y_over, y_src

    # GridSearchCV params

    param_grid_logit = [
        {
            'penalty' : ['l1', 'l2', 'elasticnet', 'none'],
            'C' : np.logspace(-4, 4, 20),
            'solver' : ['lbfgs','newton-cg','liblinear','sag','saga'],
            'max_iter' : [10, 100, 500, 1000]

        }
    ]

    param_grid_sgdc = [
        {
            'loss' : ['hinge', 'log', 'modified_huber', 'squared_hinge', 'perceptron']
            'penalty' : ['l1', 'l2', 'elasticnet'],
            'alpha' : [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000],
            'learning_rate' : ['constant', 'optimal', 'invscaling', 'adaptive'],
            'class_weight' : [{1:0.5, 0:0.5}, {1:0.4, 0:0.6}, {1:0.6, 0:0.4}, {1:0.7,
            'eta0' : [1, 10, 100]
        }
    ]

    param_grid_lda = [
        {
            'solver': ['svd', 'lsqr', 'eigen']
        }
    ]

    param_grid_qda = [
        {
            'reg_param': [0.1, 0.2, 0.3, 0.4, 0.5]
        }
    ]
```

```
param_grid_rfc = [
    {
        'n_estimators': [25, 50, 100, 150],
        'max_features': ['sqrt', 'log2', None],
        'max_depth': [3, 6, 9, 12],
        'max_leaf_nodes': [3, 6, 9, 12]
    }
]


param_grid_mlpc = [
    {
        'hidden_layer_sizes': [(10,30,10),(20,)],
        'activation': ['tanh', 'relu'],
        'solver': ['sgd', 'adam'],
        'alpha': [0.0001, 0.05],
        'learning_rate': ['constant','adaptive']
    }
]


clf_logit = GridSearchCV(LogisticRegression(), param_grid = param_grid_logit, cv =
y_pred_logit = clf_logit.predict(X_test)
cnf_matrix_logit = metrics.confusion_matrix(y_test, y_pred_logit)

clf_sgdc = GridSearchCV(SGDClassifier(), param_grid = param_grid_sgdc, cv = 3, ver
y_pred_sgdc = clf_sgdc.predict(X_test)
cnf_matrix_sgdc = metrics.confusion_matrix(y_test, y_pred_sgdc)

clf_lda = GridSearchCV(LinearDiscriminantAnalysis(), param_grid = param_grid_lda,
y_pred_lda = clf_lda.predict(X_test)
cnf_matrix_lda = metrics.confusion_matrix(y_test, y_pred_lda)

clf_qda = GridSearchCV(QuadraticDiscriminantAnalysis(), param_grid = param_grid_qd
y_pred_qda = clf_qda.predict(X_test)
cnf_matrix_qda = metrics.confusion_matrix(y_test, y_pred_qda)

clf_rfc = GridSearchCV(RandomForestClassifier(), param_grid = param_grid_rfc, cv =
y_pred_rfc = clf_rfc.predict(X_test)
cnf_matrix_rfc = metrics.confusion_matrix(y_test, y_pred_rfc)

clf_mlpc = GridSearchCV(MLPClassifier(max_iter=100), param_grid = param_grid_mlpc
y_pred_mlpc = clf_mlpc.predict(X_test)
cnf_matrix_mlpc = metrics.confusion_matrix(y_test, y_pred_mlpc)

return cnf_matrix_sgdc, cnf_matrix_logit, cnf_matrix_lda, cnf_matrix_qda, cnf_matr
```

```
def make_ci_frame(X, y):

    cnf_matrix_sgdc, cnf_matrix_logit, cnf_matrix_lda, cnf_matrix_qda, cnf_matrix_rfc

    my_frames_ = []

    for obs, name in zip([cnf_matrix_sgdc, cnf_matrix_logit, cnf_matrix_lda, cnf_matri
                          ['cnf_matrix_sgdc', 'cnf_matrix_logit', 'cnf_matrix_lda', 'cn

        TP, FP, FN, TN = explode_confusion_matrix(obs)
        acuuracy, precision, recall, f1 = return_metrics(TP, FP, FN, TN)
        success_accuracy, unsuccess_accuracy, success_precision, unsuccess_precision,
        models_frame = make_bounds(success_accuracy, unsuccess_accuracy, success_preci
        models_frame['accuracy'] = acuuracy
        models_frame['precision'] = precision
        models_frame['recall'] = recall
        models_frame['f1'] = f1
        models_frame['model'] = name
        my_frames_.append(models_frame)

    return pd.concat(my_frames_)

def make_ci_frame_over(X_over, y_over, X_src, y_src):

    cnf_matrix_sgdc, cnf_matrix_logit, cnf_matrix_lda, cnf_matrix_qda, cnf_matrix_rfc

    my_frames_ = []

    for obs, name in zip([cnf_matrix_sgdc, cnf_matrix_logit, cnf_matrix_lda, cnf_matri
                          ['cnf_matrix_sgdc', 'cnf_matrix_logit', 'cnf_matrix_lda', 'cn

        TP, FP, FN, TN = explode_confusion_matrix(obs)
        acuuracy, precision, recall, f1 = return_metrics(TP, FP, FN, TN)
        success_accuracy, unsuccess_accuracy, success_precision, unsuccess_precision,
        models_frame = make_bounds(success_accuracy, unsuccess_accuracy, success_preci
        models_frame['accuracy'] = acuuracy
        models_frame['precision'] = precision
        models_frame['recall'] = recall
        models_frame['f1'] = f1
        models_frame['model'] = name
        my_frames_.append(models_frame)

    return pd.concat(my_frames_)

def plot_confidence_interval(x, metric, upper, lower, color='#2187bb', horizontal_line

    mean = metric
    left = x - horizontal_line_width / 2
    top = upper
    right = x + horizontal_line_width / 2
```

```
    bottom = lower
    plt.plot([x, x], [top, bottom], color=color)
    plt.plot([left, right], [top, top], color=color)
    plt.plot([left, right], [bottom, bottom], color=color)
    plt.plot(x, mean, 'o', color='#f44336')

def make_plots(dataset, metric, lower, upper):

    print(metric)
    plot_confidence_interval(x = 1, metric = dataset[dataset['approximation']=='norma
    plot_confidence_interval(x = 2, metric = dataset[dataset['approximation']=='agres
    plot_confidence_interval(x = 3, metric = dataset[dataset['approximation']=='beta'
    plot_confidence_interval(x = 4, metric = dataset[dataset['approximation']=='wilso
    plot_confidence_interval(x = 5, metric = dataset[dataset['approximation']=='jeffr
    plt.xticks([1, 2, 3, 4, 5], ['normal', 'agresti_coull', 'beta', 'wilson','jeffrey
    plt.show()

def make_latex(df):
    print(df.to_latex().replace('_', '\_'))
    print('-'*40)
```

B Granular CI estimates

Table 4: DR=10%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8566	0.8839	0.9547	0.9711	0.8859	0.9111	0.9219	0.9372	0.0274	0.0163	0.0252	0.0154
1	AC	0.8559	0.8833	0.9538	0.9703	0.8852	0.9104	0.9215	0.9369	0.0274	0.0165	0.0253	0.0154
2	beta	0.8560	0.8836	0.9539	0.9706	0.8852	0.9107	0.9215	0.9370	0.0276	0.0166	0.0255	0.0155
3	wilson	0.8559	0.8833	0.9538	0.9702	0.8852	0.9104	0.9215	0.9369	0.0274	0.0164	0.0252	0.0154
4	JEF	0.8561	0.8835	0.9541	0.9704	0.8854	0.9106	0.9216	0.9370	0.0274	0.0163	0.0252	0.0154

Table 5: DR=10%, extra 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8563	0.8837	0.9544	0.9708	0.8858	0.9111	0.9217	0.9371	0.0274	0.0164	0.0252	0.0154
1	AC	0.8557	0.8831	0.9535	0.9700	0.8851	0.9104	0.9213	0.9367	0.0274	0.0166	0.0253	0.0154
2	beta	0.8557	0.8834	0.9536	0.9703	0.8852	0.9107	0.9214	0.9369	0.0276	0.0167	0.0255	0.0155
3	wilson	0.8557	0.8831	0.9535	0.9700	0.8851	0.9104	0.9213	0.9367	0.0274	0.0165	0.0252	0.0154
4	JEF	0.8559	0.8833	0.9538	0.9702	0.8854	0.9106	0.9215	0.9368	0.0274	0.0164	0.0252	0.0154

Table 6: DR=10%, less than 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8681	0.8944	0.9802	0.9906	0.8796	0.9051	0.9293	0.9438	0.0264	0.0104	0.0255	0.0145
1	AC	0.8674	0.8938	0.9791	0.9898	0.8789	0.9044	0.9289	0.9434	0.0264	0.0107	0.0255	0.0146
2	beta	0.8675	0.8941	0.9794	0.9901	0.8790	0.9047	0.9289	0.9436	0.0266	0.0107	0.0257	0.0147
3	wilson	0.8674	0.8938	0.9792	0.9897	0.8789	0.9044	0.9289	0.9434	0.0264	0.0105	0.0255	0.0145
4	JEF	0.8676	0.8940	0.9795	0.9899	0.8791	0.9046	0.9290	0.9435	0.0263	0.0104	0.0255	0.0145

Table 7: DR=10%, less than 5 features and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8681	0.8944	0.9802	0.9906	0.8796	0.9051	0.9293	0.9438	0.0264	0.0104	0.0255	0.0145
1	AC	0.8674	0.8938	0.9791	0.9898	0.8789	0.9044	0.9289	0.9434	0.0264	0.0107	0.0255	0.0146
2	beta	0.8675	0.8941	0.9794	0.9901	0.8790	0.9047	0.9289	0.9436	0.0266	0.0107	0.0257	0.0147
3	wilson	0.8674	0.8938	0.9792	0.9897	0.8789	0.9044	0.9289	0.9434	0.0264	0.0105	0.0255	0.0145
4	JEF	0.8676	0.8940	0.9795	0.9899	0.8791	0.9046	0.9290	0.9435	0.0263	0.0104	0.0255	0.0145

Table 8: Oversample DR=10% to DR=50%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8443	0.8727	0.9055	0.9292	0.9115	0.9346	0.9119	0.9285	0.0284	0.0238	0.0231	0.0166
1	AC	0.8437	0.8721	0.9046	0.9285	0.9107	0.9339	0.9115	0.9281	0.0284	0.0239	0.0232	0.0166
2	beta	0.8438	0.8724	0.9047	0.9288	0.9108	0.9342	0.9115	0.9283	0.0286	0.0241	0.0234	0.0167
3	wilson	0.8437	0.8721	0.9047	0.9285	0.9107	0.9338	0.9115	0.9281	0.0284	0.0238	0.0231	0.0166
4	JEF	0.8439	0.8723	0.9049	0.9287	0.9109	0.9340	0.9116	0.9282	0.0284	0.0238	0.0231	0.0166

Table 9: Oversample DR=10% to DR=50%, extra 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8474	0.8756	0.9106	0.9337	0.9106	0.9337	0.9139	0.9303	0.0281	0.0231	0.0231	0.0164
1	AC	0.8468	0.8750	0.9097	0.9330	0.9097	0.9330	0.9135	0.9299	0.0282	0.0232	0.0232	0.0164
2	beta	0.8469	0.8753	0.9098	0.9333	0.9098	0.9333	0.9136	0.9301	0.0284	0.0234	0.0234	0.0165
3	wilson	0.8468	0.8750	0.9097	0.9329	0.9097	0.9329	0.9135	0.9299	0.0281	0.0232	0.0232	0.0164
4	JEF	0.8470	0.8751	0.9100	0.9331	0.9100	0.9331	0.9137	0.9300	0.0281	0.0232	0.0232	0.0164

Table 10: Oversample DR=10% to DR=50%, less then 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.7423	0.7772	0.8009	0.8342	0.8897	0.9165	0.8473	0.8691	0.0348	0.0334	0.0269	0.0218
1	AC	0.7419	0.7767	0.8003	0.8336	0.8888	0.9157	0.8469	0.8688	0.0348	0.0334	0.0269	0.0218
2	beta	0.7419	0.7770	0.8003	0.8339	0.8889	0.9161	0.8470	0.8689	0.0350	0.0336	0.0272	0.0220
3	wilson	0.7419	0.7767	0.8003	0.8336	0.8888	0.9157	0.8469	0.8688	0.0348	0.0334	0.0269	0.0218
4	JEF	0.7420	0.7768	0.8004	0.8338	0.8891	0.9159	0.8470	0.8689	0.0348	0.0333	0.0269	0.0218

Table 11: Oversample DR=10% to DR=50%, less than 5 features and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.7418	0.7767	0.8006	0.8340	0.8893	0.9162	0.8470	0.8688	0.0348	0.0334	0.0269	0.0218
1	AC	0.7414	0.7762	0.8000	0.8334	0.8885	0.9155	0.8466	0.8685	0.0348	0.0334	0.0270	0.0219
2	beta	0.7414	0.7765	0.8000	0.8337	0.8886	0.9158	0.8467	0.8686	0.0351	0.0337	0.0272	0.0220
3	wilson	0.7414	0.7762	0.8000	0.8334	0.8885	0.9154	0.8466	0.8685	0.0348	0.0334	0.0269	0.0219
4	JEF	0.7415	0.7763	0.8002	0.8335	0.8888	0.9156	0.8467	0.8686	0.0348	0.0334	0.0269	0.0218

Table 12: DR=3%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9175	0.9385	0.9479	0.9648	0.9616	0.9761	0.9570	0.9681	0.0211	0.0169	0.0145	0.0112
1	AC	0.9167	0.9379	0.9470	0.9641	0.9607	0.9754	0.9566	0.9678	0.0211	0.0170	0.0146	0.0112
2	beta	0.9168	0.9381	0.9472	0.9644	0.9609	0.9756	0.9566	0.9679	0.0213	0.0172	0.0148	0.0113
3	wilson	0.9167	0.9378	0.9471	0.9640	0.9608	0.9753	0.9566	0.9677	0.0211	0.0170	0.0146	0.0112
4	JEF	0.9170	0.9380	0.9473	0.9642	0.9610	0.9755	0.9567	0.9679	0.0211	0.0169	0.0145	0.0112

Table 13: DR=3%, extra 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9175	0.9385	0.9479	0.9648	0.9616	0.9761	0.9570	0.9681	0.0211	0.0169	0.0145	0.0112
1	AC	0.9167	0.9379	0.9470	0.9641	0.9607	0.9754	0.9566	0.9678	0.0211	0.0170	0.0146	0.0112
2	beta	0.9168	0.9381	0.9472	0.9644	0.9609	0.9756	0.9566	0.9679	0.0213	0.0172	0.0148	0.0113
3	wilson	0.9167	0.9378	0.9471	0.9640	0.9608	0.9753	0.9566	0.9677	0.0211	0.0170	0.0146	0.0112
4	JEF	0.9170	0.9380	0.9473	0.9642	0.9610	0.9755	0.9567	0.9679	0.0211	0.0169	0.0145	0.0112

Table 14: DR=3%, less than 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8820	0.9070	0.9101	0.9324	0.9608	0.9757	0.9373	0.9509	0.0250	0.0223	0.0149	0.0136
1	AC	0.8813	0.9064	0.9093	0.9317	0.9599	0.9749	0.9369	0.9506	0.0251	0.0224	0.0151	0.0137
2	beta	0.8814	0.9067	0.9094	0.9320	0.9600	0.9752	0.9370	0.9507	0.0253	0.0226	0.0152	0.0137
3	wilson	0.8813	0.9064	0.9093	0.9317	0.9599	0.9749	0.9369	0.9506	0.0250	0.0223	0.0150	0.0136
4	JEF	0.8815	0.9065	0.9095	0.9318	0.9602	0.9751	0.9370	0.9507	0.0250	0.0223	0.0149	0.0136

Table 15: DR=3%, less than 5 features and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8820	0.9070	0.9101	0.9324	0.9608	0.9757	0.9373	0.9509	0.0250	0.0223	0.0149	0.0136
1	AC	0.8813	0.9064	0.9093	0.9317	0.9599	0.9749	0.9369	0.9506	0.0251	0.0224	0.0151	0.0137
2	beta	0.8814	0.9067	0.9094	0.9320	0.9600	0.9752	0.9370	0.9507	0.0253	0.0226	0.0152	0.0137
3	wilson	0.8813	0.9064	0.9093	0.9317	0.9599	0.9749	0.9369	0.9506	0.0250	0.0223	0.0150	0.0136
4	JEF	0.8815	0.9065	0.9095	0.9318	0.9602	0.9751	0.9370	0.9507	0.0250	0.0223	0.0149	0.0136

Table 16: Oversample DR=3% to DR=50%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9010	0.9240	0.9236	0.9442	0.9680	0.9813	0.9476	0.9600	0.0230	0.0206	0.0133	0.0124
1	AC	0.9003	0.9234	0.9228	0.9435	0.9671	0.9806	0.9472	0.9597	0.0231	0.0207	0.0135	0.0125
2	beta	0.9004	0.9236	0.9229	0.9437	0.9672	0.9808	0.9473	0.9598	0.0233	0.0208	0.0136	0.0126
3	wilson	0.9003	0.9233	0.9228	0.9434	0.9671	0.9805	0.9472	0.9596	0.0230	0.0206	0.0134	0.0124
4	JEF	0.9005	0.9235	0.9230	0.9436	0.9674	0.9807	0.9473	0.9598	0.0230	0.0206	0.0133	0.0124

Table 17: Oversample DR=3% to DR=10%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9103	0.9322	0.9383	0.9568	0.9633	0.9775	0.9530	0.9647	0.0219	0.0185	0.0142	0.0117
1	AC	0.9095	0.9316	0.9375	0.9561	0.9624	0.9767	0.9526	0.9643	0.0220	0.0186	0.0144	0.0118
2	beta	0.9096	0.9318	0.9376	0.9564	0.9625	0.9770	0.9526	0.9645	0.0222	0.0187	0.0145	0.0118
3	wilson	0.9096	0.9315	0.9375	0.9560	0.9624	0.9767	0.9526	0.9643	0.0220	0.0185	0.0143	0.0117
4	JEF	0.9098	0.9317	0.9378	0.9562	0.9627	0.9769	0.9527	0.9644	0.0219	0.0185	0.0142	0.0117

Table 18: Oversample DR=3% to DR=5%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9140	0.9355	0.9423	0.9601	0.9634	0.9776	0.9550	0.9664	0.0215	0.0178	0.0142	0.0114
1	AC	0.9133	0.9348	0.9414	0.9594	0.9625	0.9768	0.9546	0.9661	0.0216	0.0180	0.0143	0.0115
2	beta	0.9134	0.9351	0.9416	0.9597	0.9627	0.9771	0.9547	0.9662	0.0217	0.0181	0.0144	0.0116
3	wilson	0.9133	0.9348	0.9415	0.9594	0.9625	0.9768	0.9546	0.9661	0.0215	0.0179	0.0142	0.0115
4	JEF	0.9135	0.9350	0.9417	0.9595	0.9628	0.9770	0.9547	0.9662	0.0215	0.0179	0.0142	0.0114

Table 19: Oversample DR=3% to DR=50%, and 5 extra features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9007	0.9238	0.9236	0.9442	0.9677	0.9811	0.9475	0.9599	0.0230	0.0206	0.0134	0.0124
1	AC	0.9000	0.9231	0.9228	0.9435	0.9668	0.9803	0.9471	0.9595	0.0231	0.0207	0.0136	0.0125
2	beta	0.9001	0.9234	0.9229	0.9437	0.9669	0.9806	0.9471	0.9597	0.0233	0.0208	0.0137	0.0126
3	wilson	0.9000	0.9231	0.9228	0.9434	0.9668	0.9803	0.9471	0.9595	0.0231	0.0206	0.0135	0.0125
4	JEF	0.9002	0.9233	0.9230	0.9436	0.9671	0.9805	0.9472	0.9596	0.0230	0.0206	0.0134	0.0124

Table 20: Oversample DR=3% to DR=10%, and 5 extra features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9103	0.9322	0.9383	0.9568	0.9633	0.9775	0.9530	0.9647	0.0219	0.0185	0.0142	0.0117
1	AC	0.9095	0.9316	0.9375	0.9561	0.9624	0.9767	0.9526	0.9643	0.0220	0.0186	0.0144	0.0118
2	beta	0.9096	0.9318	0.9376	0.9564	0.9625	0.9770	0.9526	0.9645	0.0222	0.0187	0.0145	0.0118
3	wilson	0.9096	0.9315	0.9375	0.9560	0.9624	0.9767	0.9526	0.9643	0.0220	0.0185	0.0143	0.0117
4	JEF	0.9098	0.9317	0.9378	0.9562	0.9627	0.9769	0.9527	0.9644	0.0219	0.0185	0.0142	0.0117

Table 21: Oversample DR=3% to DR=5%, and 5 extra features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9140	0.9355	0.9423	0.9601	0.9634	0.9776	0.9550	0.9664	0.0215	0.0178	0.0142	0.0114
1	AC	0.9133	0.9348	0.9414	0.9594	0.9625	0.9768	0.9546	0.9661	0.0216	0.0180	0.0143	0.0115
2	beta	0.9134	0.9351	0.9416	0.9597	0.9627	0.9771	0.9547	0.9662	0.0217	0.0181	0.0144	0.0116
3	wilson	0.9133	0.9348	0.9415	0.9594	0.9625	0.9768	0.9546	0.9661	0.0215	0.0179	0.0142	0.0115
4	JEF	0.9135	0.9350	0.9417	0.9595	0.9628	0.9770	0.9547	0.9662	0.0215	0.0179	0.0142	0.0114

Table 22: Oversample DR=3% to DR=50%, less than 5 featuress

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8946	0.9184	0.9225	0.9432	0.9621	0.9767	0.9444	0.9572	0.0237	0.0207	0.0145	0.0128
1	AC	0.8939	0.9177	0.9217	0.9425	0.9612	0.9759	0.9440	0.9568	0.0238	0.0208	0.0147	0.0128
2	beta	0.8940	0.9180	0.9218	0.9428	0.9614	0.9762	0.9440	0.9569	0.0240	0.0210	0.0148	0.0129
3	wilson	0.8940	0.9177	0.9217	0.9425	0.9613	0.9759	0.9440	0.9568	0.0237	0.0208	0.0146	0.0128
4	JEF	0.8942	0.9179	0.9219	0.9427	0.9615	0.9761	0.9441	0.9569	0.0237	0.0207	0.0146	0.0128

Table 23: Oversample DR=3% to DR=10%, less than 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8728	0.8987	0.8980	0.9217	0.9630	0.9775	0.9320	0.9462	0.0259	0.0237	0.0145	0.0142
1	AC	0.8721	0.8981	0.8973	0.9210	0.9620	0.9767	0.9316	0.9458	0.0260	0.0238	0.0147	0.0143
2	beta	0.8722	0.8984	0.8973	0.9213	0.9622	0.9770	0.9316	0.9460	0.0262	0.0240	0.0148	0.0144
3	wilson	0.8722	0.8981	0.8973	0.9210	0.9621	0.9767	0.9316	0.9458	0.0259	0.0237	0.0146	0.0142
4	JEF	0.8723	0.8982	0.8975	0.9212	0.9624	0.9769	0.9317	0.9459	0.0259	0.0237	0.0145	0.0142

Table 24: Oversample DR=3% to DR=5%, less than 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8801	0.9054	0.9062	0.9290	0.9627	0.9772	0.9362	0.9499	0.0252	0.0228	0.0145	0.0138
1	AC	0.8795	0.9047	0.9055	0.9283	0.9618	0.9765	0.9358	0.9496	0.0253	0.0228	0.0147	0.0138
2	beta	0.8796	0.9050	0.9056	0.9286	0.9619	0.9767	0.9358	0.9497	0.0255	0.0230	0.0148	0.0139
3	wilson	0.8795	0.9047	0.9055	0.9283	0.9618	0.9764	0.9358	0.9495	0.0252	0.0228	0.0146	0.0138
4	JEF	0.8797	0.9049	0.9057	0.9285	0.9621	0.9766	0.9359	0.9497	0.0252	0.0228	0.0146	0.0138

Table 25: Oversample DR=3% to DR=50%, less than 5 featuress and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9204	0.9411	0.9504	0.9669	0.9623	0.9766	0.9586	0.9695	0.0207	0.0165	0.0143	0.0109
1	AC	0.9197	0.9404	0.9496	0.9662	0.9614	0.9759	0.9581	0.9691	0.0208	0.0166	0.0145	0.0110
2	beta	0.9198	0.9407	0.9497	0.9665	0.9615	0.9761	0.9582	0.9693	0.0209	0.0168	0.0146	0.0111
3	wilson	0.9197	0.9404	0.9496	0.9662	0.9614	0.9758	0.9581	0.9691	0.0207	0.0165	0.0144	0.0110
4	JEF	0.9199	0.9406	0.9499	0.9664	0.9617	0.9760	0.9583	0.9692	0.0207	0.0165	0.0144	0.0109

Table 26: Oversample DR=3% to DR=10%, less than 5 features and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8725	0.8985	0.8977	0.9215	0.9630	0.9775	0.9318	0.9461	0.0259	0.0237	0.0145	0.0142
1	AC	0.8719	0.8979	0.8970	0.9208	0.9620	0.9767	0.9314	0.9457	0.0260	0.0238	0.0147	0.0143
2	beta	0.8719	0.8981	0.8971	0.9211	0.9622	0.9770	0.9315	0.9458	0.0262	0.0240	0.0148	0.0144
3	wilson	0.8719	0.8978	0.8970	0.9208	0.9621	0.9767	0.9314	0.9457	0.0259	0.0238	0.0146	0.0143
4	JEF	0.8721	0.8980	0.8972	0.9210	0.9623	0.9769	0.9315	0.9458	0.0259	0.0237	0.0146	0.0142

Table 27: Oversample DR=3% to DR=5%, less than 5 features and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8804	0.9056	0.9065	0.9292	0.9627	0.9772	0.9363	0.9501	0.0252	0.0227	0.0145	0.0137
1	AC	0.8797	0.9050	0.9057	0.9285	0.9618	0.9765	0.9359	0.9497	0.0252	0.0228	0.0147	0.0138
2	beta	0.8798	0.9052	0.9058	0.9288	0.9619	0.9768	0.9360	0.9498	0.0254	0.0230	0.0148	0.0139
3	wilson	0.8798	0.9049	0.9058	0.9285	0.9618	0.9764	0.9359	0.9497	0.0252	0.0228	0.0146	0.0138
4	JEF	0.8799	0.9051	0.9060	0.9287	0.9621	0.9766	0.9360	0.9498	0.0252	0.0227	0.0145	0.0137

Table 28: DR=0.1%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9934	0.9986	0.9973	1.0000	0.9951	0.9994	0.9967	0.9993	0.0051	0.0027	0.0043	0.0026
1	AC	0.9924	0.9980	0.9961	0.9998	0.9940	0.9989	0.9962	0.9990	0.0056	0.0037	0.0049	0.0028
2	beta	0.9926	0.9981	0.9965	0.9997	0.9943	0.9989	0.9963	0.9991	0.0055	0.0033	0.0046	0.0027
3	wilson	0.9925	0.9979	0.9963	0.9996	0.9941	0.9987	0.9962	0.9989	0.0054	0.0033	0.0046	0.0027
4	JEF	0.9928	0.9980	0.9966	0.9997	0.9945	0.9988	0.9964	0.9990	0.0052	0.0030	0.0044	0.0026

Table 29: DR=0.1%, extra 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9934	0.9986	0.9973	1.0000	0.9951	0.9994	0.9967	0.9993	0.0051	0.0027	0.0043	0.0026
1	AC	0.9924	0.9980	0.9961	0.9998	0.9940	0.9989	0.9962	0.9990	0.0056	0.0037	0.0049	0.0028
2	beta	0.9926	0.9981	0.9965	0.9997	0.9943	0.9989	0.9963	0.9991	0.0055	0.0033	0.0046	0.0027
3	wilson	0.9925	0.9979	0.9963	0.9996	0.9941	0.9987	0.9962	0.9989	0.0054	0.0033	0.0046	0.0027
4	JEF	0.9928	0.9980	0.9966	0.9997	0.9945	0.9988	0.9964	0.9990	0.0052	0.0030	0.0044	0.0026

Table 30: DR=0.1%, less than 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9853	0.9937	0.9886	0.9958	0.9951	0.9994	0.9926	0.9968	0.0083	0.0072	0.0043	0.0042
1	AC	0.9844	0.9930	0.9877	0.9952	0.9940	0.9988	0.9921	0.9965	0.0086	0.0075	0.0049	0.0043
2	beta	0.9846	0.9932	0.9879	0.9953	0.9943	0.9989	0.9922	0.9966	0.0086	0.0075	0.0046	0.0043
3	wilson	0.9845	0.9929	0.9877	0.9951	0.9941	0.9987	0.9922	0.9964	0.0085	0.0073	0.0046	0.0043
4	JEF	0.9847	0.9931	0.9880	0.9953	0.9944	0.9988	0.9923	0.9965	0.0083	0.0072	0.0044	0.0042

Table 31: DR=0.1%, less than 5 features and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9853	0.9937	0.9886	0.9958	0.9951	0.9994	0.9926	0.9968	0.0083	0.0072	0.0043	0.0042
1	AC	0.9844	0.9930	0.9877	0.9952	0.9940	0.9988	0.9921	0.9965	0.0086	0.0075	0.0049	0.0043
2	beta	0.9846	0.9932	0.9879	0.9953	0.9943	0.9989	0.9922	0.9966	0.0086	0.0075	0.0046	0.0043
3	wilson	0.9845	0.9929	0.9877	0.9951	0.9941	0.9987	0.9922	0.9964	0.0085	0.0073	0.0046	0.0043
4	JEF	0.9847	0.9931	0.9880	0.9953	0.9944	0.9988	0.9923	0.9965	0.0083	0.0072	0.0044	0.0042

Table 32: Oversample DR = 0.1% to DR=10%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9934	0.9986	0.9973	1.0000	0.9951	0.9994	0.9967	0.9993	0.0051	0.0027	0.0043	0.0026
1	AC	0.9924	0.9980	0.9961	0.9998	0.9940	0.9989	0.9962	0.9990	0.0056	0.0037	0.0049	0.0028
2	beta	0.9926	0.9981	0.9965	0.9997	0.9943	0.9989	0.9963	0.9991	0.0055	0.0033	0.0046	0.0027
3	wilson	0.9925	0.9979	0.9963	0.9996	0.9941	0.9987	0.9962	0.9989	0.0054	0.0033	0.0046	0.0027
4	JEF	0.9928	0.9980	0.9966	0.9997	0.9945	0.9988	0.9964	0.9990	0.0052	0.0030	0.0044	0.0026

Table 33: Oversample DR = 0.1% to DR=5%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9938	0.9987	0.9977	1.0000	0.9951	0.9994	0.9969	0.9994	0.0050	0.0023	0.0043	0.0025
1	AC	0.9927	0.9982	0.9964	0.9999	0.9940	0.9989	0.9963	0.9991	0.0055	0.0035	0.0049	0.0028
2	beta	0.9930	0.9983	0.9968	0.9998	0.9943	0.9989	0.9965	0.9991	0.0053	0.0030	0.0046	0.0027
3	wilson	0.9928	0.9980	0.9966	0.9997	0.9941	0.9987	0.9964	0.9990	0.0052	0.0031	0.0046	0.0026
4	JEF	0.9931	0.9982	0.9970	0.9998	0.9945	0.9988	0.9966	0.9991	0.0051	0.0027	0.0044	0.0025

Table 34: Oversample DR = 0.1% to DR=3%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9934	0.9986	0.9973	1.0000	0.9951	0.9994	0.9967	0.9993	0.0051	0.0027	0.0043	0.0026
1	AC	0.9924	0.9980	0.9961	0.9998	0.9940	0.9989	0.9962	0.9990	0.0056	0.0037	0.0049	0.0028
2	beta	0.9926	0.9981	0.9965	0.9997	0.9943	0.9989	0.9963	0.9991	0.0055	0.0033	0.0046	0.0027
3	wilson	0.9925	0.9979	0.9963	0.9996	0.9941	0.9987	0.9962	0.9989	0.0054	0.0033	0.0046	0.0027
4	JEF	0.9928	0.9980	0.9966	0.9997	0.9945	0.9988	0.9964	0.9990	0.0052	0.0030	0.0044	0.0026

Table 35: Oversample DR = 0.1% to DR=1%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9934	0.9986	0.9973	1.0000	0.9951	0.9994	0.9967	0.9993	0.0051	0.0027	0.0043	0.0026
1	AC	0.9924	0.9980	0.9961	0.9998	0.9940	0.9989	0.9962	0.9990	0.0056	0.0037	0.0049	0.0028
2	beta	0.9926	0.9981	0.9965	0.9997	0.9943	0.9989	0.9963	0.9991	0.0055	0.0033	0.0046	0.0027
3	wilson	0.9925	0.9979	0.9963	0.9996	0.9941	0.9987	0.9962	0.9989	0.0054	0.0033	0.0046	0.0027
4	JEF	0.9928	0.9980	0.9966	0.9997	0.9945	0.9988	0.9964	0.9990	0.0052	0.0030	0.0044	0.0026

Table 36: Oversample DR = 0.1% to DR=0.5%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9934	0.9986	0.9973	1.0000	0.9951	0.9994	0.9967	0.9993	0.0051	0.0027	0.0043	0.0026
1	AC	0.9924	0.9980	0.9961	0.9998	0.9940	0.9989	0.9962	0.9990	0.0056	0.0037	0.0049	0.0028
2	beta	0.9926	0.9981	0.9965	0.9997	0.9943	0.9989	0.9963	0.9991	0.0055	0.0033	0.0046	0.0027
3	wilson	0.9925	0.9979	0.9963	0.9996	0.9941	0.9987	0.9962	0.9989	0.0054	0.0033	0.0046	0.0027
4	JEF	0.9928	0.9980	0.9966	0.9997	0.9945	0.9988	0.9964	0.9990	0.0052	0.0030	0.0044	0.0026

Table 37: Oversample DR=0.1% to DR=50%, all features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8915	0.9155	0.8936	0.9174	0.9954	0.9996	0.9428	0.9558	0.0241	0.0239	0.0043	0.0130
1	AC	0.8908	0.9149	0.8929	0.9168	0.9941	0.9991	0.9424	0.9554	0.0241	0.0239	0.0050	0.0130
2	beta	0.8909	0.9152	0.8929	0.9171	0.9945	0.9991	0.9424	0.9555	0.0243	0.0241	0.0046	0.0131
3	wilson	0.8908	0.9149	0.8929	0.9168	0.9943	0.9989	0.9424	0.9554	0.0241	0.0239	0.0046	0.0130
4	JEF	0.8910	0.9150	0.8931	0.9169	0.9947	0.9991	0.9425	0.9555	0.0241	0.0239	0.0044	0.0130

Table 38: Oversample DR=0.1% to DR=50%, and 5 extra features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8762	0.9018	0.8782	0.9037	0.9953	0.9996	0.9342	0.9482	0.0256	0.0254	0.0043	0.0139
1	AC	0.8755	0.9012	0.8776	0.9030	0.9940	0.9991	0.9338	0.9478	0.0256	0.0255	0.0051	0.0140
2	beta	0.8756	0.9014	0.8776	0.9033	0.9944	0.9991	0.9339	0.9480	0.0258	0.0257	0.0047	0.0141
3	wilson	0.8756	0.9012	0.8776	0.9030	0.9942	0.9989	0.9338	0.9478	0.0256	0.0254	0.0047	0.0140
4	JEF	0.8757	0.9013	0.8778	0.9032	0.9946	0.9990	0.9340	0.9479	0.0256	0.0254	0.0044	0.0139

Table 39: Oversample DR=0.1% to DR=50%, less than 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9862	0.9943	0.9893	0.9962	0.9954	0.9995	0.9931	0.9971	0.0080	0.0069	0.0041	0.0040
1	AC	0.9853	0.9936	0.9883	0.9956	0.9943	0.9990	0.9926	0.9968	0.0083	0.0073	0.0047	0.0042
2	beta	0.9855	0.9938	0.9885	0.9957	0.9946	0.9991	0.9927	0.9969	0.0083	0.0072	0.0044	0.0042
3	wilson	0.9854	0.9935	0.9884	0.9955	0.9944	0.9989	0.9926	0.9967	0.0082	0.0071	0.0044	0.0041
4	JEF	0.9856	0.9937	0.9887	0.9956	0.9948	0.9990	0.9928	0.9968	0.0080	0.0070	0.0042	0.0041

Table 40: Oversample DR=0.1% to DR=50%, less than 5 features and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.8607	0.8878	0.8630	0.8898	0.9948	0.9995	0.9254	0.9403	0.0270	0.0268	0.0046	0.0149
1	AC	0.8601	0.8872	0.8623	0.8892	0.9935	0.9989	0.9250	0.9400	0.0270	0.0269	0.0053	0.0149
2	beta	0.8602	0.8874	0.8624	0.8895	0.9939	0.9989	0.9251	0.9401	0.0273	0.0271	0.0050	0.0150
3	wilson	0.8601	0.8871	0.8624	0.8892	0.9937	0.9987	0.9251	0.9400	0.0270	0.0269	0.0050	0.0149
4	JEF	0.8603	0.8873	0.8625	0.8894	0.9941	0.9989	0.9252	0.9401	0.0270	0.0268	0.0047	0.0149

Table 41: Oversample DR = 0.1% to DR=10%, and 5 extra features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9934	0.9986	0.9973	1.0000	0.9951	0.9994	0.9967	0.9993	0.0051	0.0027	0.0043	0.0026
1	AC	0.9924	0.9980	0.9961	0.9998	0.9940	0.9989	0.9962	0.9990	0.0056	0.0037	0.0049	0.0028
2	beta	0.9926	0.9981	0.9965	0.9997	0.9943	0.9989	0.9963	0.9991	0.0055	0.0033	0.0046	0.0027
3	wilson	0.9925	0.9979	0.9963	0.9996	0.9941	0.9987	0.9962	0.9989	0.0054	0.0033	0.0046	0.0027
4	JEF	0.9928	0.9980	0.9966	0.9997	0.9945	0.9988	0.9964	0.9990	0.0052	0.0030	0.0044	0.0026

Table 42: Oversample DR = 0.1% to DR=5%, and 5 extra features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9938	0.9987	0.9977	1.0000	0.9951	0.9994	0.9969	0.9994	0.0050	0.0023	0.0043	0.0025
1	AC	0.9927	0.9982	0.9964	0.9999	0.9940	0.9989	0.9963	0.9991	0.0055	0.0035	0.0049	0.0028
2	beta	0.9930	0.9983	0.9968	0.9998	0.9943	0.9989	0.9965	0.9991	0.0053	0.0030	0.0046	0.0027
3	wilson	0.9928	0.9980	0.9966	0.9997	0.9941	0.9987	0.9964	0.9990	0.0052	0.0031	0.0046	0.0026
4	JEF	0.9931	0.9982	0.9970	0.9998	0.9945	0.9988	0.9966	0.9991	0.0051	0.0027	0.0044	0.0025

Table 43: Oversample DR = 0.1% to DR=3%, and 5 extra features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9934	0.9986	0.9973	1.0000	0.9951	0.9994	0.9967	0.9993	0.0051	0.0027	0.0043	0.0026
1	AC	0.9924	0.9980	0.9961	0.9998	0.9940	0.9989	0.9962	0.9990	0.0056	0.0037	0.0049	0.0028
2	beta	0.9926	0.9981	0.9965	0.9997	0.9943	0.9989	0.9963	0.9991	0.0055	0.0033	0.0046	0.0027
3	wilson	0.9925	0.9979	0.9963	0.9996	0.9941	0.9987	0.9962	0.9989	0.0054	0.0033	0.0046	0.0027
4	JEF	0.9928	0.9980	0.9966	0.9997	0.9945	0.9988	0.9964	0.9990	0.0052	0.0030	0.0044	0.0026

Table 44: Oversample DR = 0.1% to DR=1%, and 5 extra features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9934	0.9986	0.9973	1.0000	0.9951	0.9994	0.9967	0.9993	0.0051	0.0027	0.0043	0.0026
1	AC	0.9924	0.9980	0.9961	0.9998	0.9940	0.9989	0.9962	0.9990	0.0056	0.0037	0.0049	0.0028
2	beta	0.9926	0.9981	0.9965	0.9997	0.9943	0.9989	0.9963	0.9991	0.0055	0.0033	0.0046	0.0027
3	wilson	0.9925	0.9979	0.9963	0.9996	0.9941	0.9987	0.9962	0.9989	0.0054	0.0033	0.0046	0.0027
4	JEF	0.9928	0.9980	0.9966	0.9997	0.9945	0.9988	0.9964	0.9990	0.0052	0.0030	0.0044	0.0026

Table 45: Oversample DR = 0.1% to DR=0.5%, and 5 extra features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9934	0.9986	0.9973	1.0000	0.9951	0.9994	0.9967	0.9993	0.0051	0.0027	0.0043	0.0026
1	AC	0.9924	0.9980	0.9961	0.9998	0.9940	0.9989	0.9962	0.9990	0.0056	0.0037	0.0049	0.0028
2	beta	0.9926	0.9981	0.9965	0.9997	0.9943	0.9989	0.9963	0.9991	0.0055	0.0033	0.0046	0.0027
3	wilson	0.9925	0.9979	0.9963	0.9996	0.9941	0.9987	0.9962	0.9989	0.0054	0.0033	0.0046	0.0027
4	JEF	0.9928	0.9980	0.9966	0.9997	0.9945	0.9988	0.9964	0.9990	0.0052	0.0030	0.0044	0.0026

Table 46: Oversample DR = 0.1% to DR=0.1%, and 5 extra features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9934	0.9986	0.9973	1.0000	0.9951	0.9994	0.9967	0.9993	0.0051	0.0027	0.0043	0.0026
1	AC	0.9924	0.9980	0.9961	0.9998	0.9940	0.9989	0.9962	0.9990	0.0056	0.0037	0.0049	0.0028
2	beta	0.9926	0.9981	0.9965	0.9997	0.9943	0.9989	0.9963	0.9991	0.0055	0.0033	0.0046	0.0027
3	wilson	0.9925	0.9979	0.9963	0.9996	0.9941	0.9987	0.9962	0.9989	0.0054	0.0033	0.0046	0.0027
4	JEF	0.9928	0.9980	0.9966	0.9997	0.9945	0.9988	0.9964	0.9990	0.0052	0.0030	0.0044	0.0026

Table 47: Oversample DR = 0.1% to DR=10%, less than 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9856	0.9939	0.9890	0.9960	0.9951	0.9994	0.9928	0.9969	0.0082	0.0070	0.0043	0.0041
1	AC	0.9847	0.9932	0.9880	0.9954	0.9940	0.9988	0.9923	0.9966	0.0085	0.0074	0.0049	0.0043
2	beta	0.9849	0.9934	0.9882	0.9955	0.9943	0.9989	0.9924	0.9967	0.0085	0.0073	0.0046	0.0043
3	wilson	0.9848	0.9931	0.9880	0.9953	0.9941	0.9987	0.9923	0.9965	0.0084	0.0072	0.0046	0.0042
4	JEF	0.9850	0.9933	0.9883	0.9954	0.9944	0.9988	0.9925	0.9966	0.0082	0.0071	0.0044	0.0042

Table 48: Oversample DR = 0.1% to DR=5%, less than 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9862	0.9943	0.9896	0.9964	0.9951	0.9994	0.9931	0.9971	0.0080	0.0068	0.0043	0.0040
1	AC	0.9853	0.9936	0.9886	0.9958	0.9940	0.9988	0.9926	0.9968	0.0083	0.0072	0.0049	0.0042
2	beta	0.9855	0.9938	0.9888	0.9959	0.9943	0.9989	0.9927	0.9969	0.0083	0.0071	0.0046	0.0042
3	wilson	0.9854	0.9935	0.9887	0.9957	0.9941	0.9987	0.9926	0.9967	0.0082	0.0070	0.0046	0.0041
4	JEF	0.9856	0.9937	0.9890	0.9958	0.9944	0.9988	0.9928	0.9968	0.0080	0.0069	0.0044	0.0041

Table 49: Oversample DR = 0.1% to DR=3%, less than 5 featuress

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9862	0.9943	0.9896	0.9964	0.9951	0.9994	0.9931	0.9971	0.0080	0.0068	0.0043	0.0040
1	AC	0.9853	0.9936	0.9886	0.9958	0.9940	0.9988	0.9926	0.9968	0.0083	0.0072	0.0049	0.0042
2	beta	0.9855	0.9938	0.9888	0.9959	0.9943	0.9989	0.9927	0.9969	0.0083	0.0071	0.0046	0.0042
3	wilson	0.9854	0.9935	0.9887	0.9957	0.9941	0.9987	0.9926	0.9967	0.0082	0.0070	0.0046	0.0041
4	JEF	0.9856	0.9937	0.9890	0.9958	0.9944	0.9988	0.9928	0.9968	0.0080	0.0069	0.0044	0.0041

Table 50: Oversample DR = 0.1% to DR=1%, less than 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9859	0.9941	0.9893	0.9962	0.9951	0.9994	0.9929	0.9970	0.0081	0.0069	0.0043	0.0041
1	AC	0.9850	0.9934	0.9883	0.9956	0.9940	0.9988	0.9924	0.9967	0.0084	0.0073	0.0049	0.0042
2	beta	0.9852	0.9936	0.9885	0.9957	0.9943	0.9989	0.9926	0.9968	0.0084	0.0072	0.0046	0.0042
3	wilson	0.9851	0.9933	0.9884	0.9955	0.9941	0.9987	0.9925	0.9966	0.0083	0.0071	0.0046	0.0042
4	JEF	0.9853	0.9935	0.9887	0.9956	0.9944	0.9988	0.9926	0.9967	0.0081	0.0070	0.0044	0.0041

Table 51: Oversample DR = 0.1% to DR=0.5%, less than 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9859	0.9941	0.9893	0.9962	0.9951	0.9994	0.9929	0.9970	0.0081	0.0069	0.0043	0.0041
1	AC	0.9850	0.9934	0.9883	0.9956	0.9940	0.9988	0.9924	0.9967	0.0084	0.0073	0.0049	0.0042
2	beta	0.9852	0.9936	0.9885	0.9957	0.9943	0.9989	0.9926	0.9968	0.0084	0.0072	0.0046	0.0042
3	wilson	0.9851	0.9933	0.9884	0.9955	0.9941	0.9987	0.9925	0.9966	0.0083	0.0071	0.0046	0.0042
4	JEF	0.9853	0.9935	0.9887	0.9956	0.9944	0.9988	0.9926	0.9967	0.0081	0.0070	0.0044	0.0041

Table 52: Oversample DR = 0.1% to DR=0.1%, less than 5 features

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9853	0.9937	0.9886	0.9958	0.9951	0.9994	0.9926	0.9968	0.0083	0.0072	0.0043	0.0042
1	AC	0.9844	0.9930	0.9877	0.9952	0.9940	0.9988	0.9921	0.9965	0.0086	0.0075	0.0049	0.0043
2	beta	0.9846	0.9932	0.9879	0.9953	0.9943	0.9989	0.9922	0.9966	0.0086	0.0075	0.0046	0.0043
3	wilson	0.9845	0.9929	0.9877	0.9951	0.9941	0.9987	0.9922	0.9964	0.0085	0.0073	0.0046	0.0043
4	JEF	0.9847	0.9931	0.9880	0.9953	0.9944	0.9988	0.9923	0.9965	0.0083	0.0072	0.0044	0.0042

Table 53: Oversample DR = 0.1% to DR=10%, less than 5 features and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9851	0.9934	0.9883	0.9956	0.9951	0.9994	0.9925	0.9967	0.0084	0.0073	0.0043	0.0042
1	AC	0.9841	0.9928	0.9873	0.9950	0.9940	0.9988	0.9920	0.9964	0.0087	0.0076	0.0049	0.0044
2	beta	0.9843	0.9930	0.9876	0.9952	0.9943	0.9989	0.9921	0.9965	0.0087	0.0076	0.0046	0.0044
3	wilson	0.9842	0.9927	0.9874	0.9949	0.9941	0.9987	0.9920	0.9963	0.0085	0.0075	0.0046	0.0043
4	JEF	0.9844	0.9929	0.9877	0.9951	0.9944	0.9988	0.9922	0.9964	0.0084	0.0073	0.0044	0.0043

Table 54: Oversample DR = 0.1% to DR=5%, less than 5 features and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9862	0.9943	0.9896	0.9964	0.9951	0.9994	0.9931	0.9971	0.0080	0.0068	0.0043	0.0040
1	AC	0.9853	0.9936	0.9886	0.9958	0.9940	0.9988	0.9926	0.9968	0.0083	0.0072	0.0049	0.0042
2	beta	0.9855	0.9938	0.9888	0.9959	0.9943	0.9989	0.9927	0.9969	0.0083	0.0071	0.0046	0.0042
3	wilson	0.9854	0.9935	0.9887	0.9957	0.9941	0.9987	0.9926	0.9967	0.0082	0.0070	0.0046	0.0041
4	JEF	0.9856	0.9937	0.9890	0.9958	0.9944	0.9988	0.9928	0.9968	0.0080	0.0069	0.0044	0.0041

Table 55: Oversample DR = 0.1% to DR=3%, less than 5 featuress and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9862	0.9943	0.9896	0.9964	0.9951	0.9994	0.9931	0.9971	0.0080	0.0068	0.0043	0.0040
1	AC	0.9853	0.9936	0.9886	0.9958	0.9940	0.9988	0.9926	0.9968	0.0083	0.0072	0.0049	0.0042
2	beta	0.9855	0.9938	0.9888	0.9959	0.9943	0.9989	0.9927	0.9969	0.0083	0.0071	0.0046	0.0042
3	wilson	0.9854	0.9935	0.9887	0.9957	0.9941	0.9987	0.9926	0.9967	0.0082	0.0070	0.0046	0.0041
4	JEF	0.9856	0.9937	0.9890	0.9958	0.9944	0.9988	0.9928	0.9968	0.0080	0.0069	0.0044	0.0041

Table 56: Oversample DR = 0.1% to DR=1%, less than 5 features and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9859	0.9941	0.9893	0.9962	0.9951	0.9994	0.9929	0.9970	0.0081	0.0069	0.0043	0.0041
1	AC	0.9850	0.9934	0.9883	0.9956	0.9940	0.9988	0.9924	0.9967	0.0084	0.0073	0.0049	0.0042
2	beta	0.9852	0.9936	0.9885	0.9957	0.9943	0.9989	0.9926	0.9968	0.0084	0.0072	0.0046	0.0042
3	wilson	0.9851	0.9933	0.9884	0.9955	0.9941	0.9987	0.9925	0.9966	0.0083	0.0071	0.0046	0.0042
4	JEF	0.9853	0.9935	0.9887	0.9956	0.9944	0.9988	0.9926	0.9967	0.0081	0.0070	0.0044	0.0041

Table 57: Oversample DR = 0.1% to DR=0.5%, less than 5 features and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9859	0.9941	0.9893	0.9962	0.9951	0.9994	0.9929	0.9970	0.0081	0.0069	0.0043	0.0041
1	AC	0.9850	0.9934	0.9883	0.9956	0.9940	0.9988	0.9924	0.9967	0.0084	0.0073	0.0049	0.0042
2	beta	0.9852	0.9936	0.9885	0.9957	0.9943	0.9989	0.9926	0.9968	0.0084	0.0072	0.0046	0.0042
3	wilson	0.9851	0.9933	0.9884	0.9955	0.9941	0.9987	0.9925	0.9966	0.0083	0.0071	0.0046	0.0042
4	JEF	0.9853	0.9935	0.9887	0.9956	0.9944	0.9988	0.9926	0.9967	0.0081	0.0070	0.0044	0.0041

Table 58: Oversample DR = 0.1% to DR=0.1%, less than 5 features and 5 extra

	CI type	accur_L	accur_U	prec_L	prec_U	recall_L	recall_U	f1_L	f1_U	accur_W	prec_W	recall_W	f1_W
0	normal	0.9853	0.9937	0.9886	0.9958	0.9951	0.9994	0.9926	0.9968	0.0083	0.0072	0.0043	0.0042
1	AC	0.9844	0.9930	0.9877	0.9952	0.9940	0.9988	0.9921	0.9965	0.0086	0.0075	0.0049	0.0043
2	beta	0.9846	0.9932	0.9879	0.9953	0.9943	0.9989	0.9922	0.9966	0.0086	0.0075	0.0046	0.0043
3	wilson	0.9845	0.9929	0.9877	0.9951	0.9941	0.9987	0.9922	0.9964	0.0085	0.0073	0.0046	0.0043
4	JEF	0.9847	0.9931	0.9880	0.9953	0.9944	0.9988	0.9923	0.9965	0.0083	0.0072	0.0044	0.0042