



[Click to Take the FREE Imbalanced Classification Crash-Course](#)

Search...



8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset

by Jason Brownlee on [August 19, 2015](#) in [Imbalanced Classification](#)



Tweet



Share



Share

Last Updated on August 15, 2020

Has this happened to you?

You are working on your dataset. You create a classification model and get 90% accuracy immediately. “*Fantastic*” you think. You dive a little deeper and discover that 90% of the data belongs to one class. Damn!

This is an example of an imbalanced dataset and the frustrating results it can cause.

In this post you will discover the tactics that you can use to deliver great results on machine learning datasets with imbalanced data.

Kick-start your project with my new book [Imbalanced Classification with Python](#), including *step-by-step tutorials* and the *Python source code* files for all examples.

Let’s get started.



Find some balance in your machine learning.
Photo by MichaEli, some rights reserved.

Coming To Grips With Imbalanced Data

I get emails about class imbalance all the time, for example:

“ I have a binary classification problem and one class is present with 60:1 ratio in my training set. I used the logistic regression and the result seems to just ignores one class.

And this:

“ I am working on a classification model. In my dataset I have three different labels to be classified, let them be A, B and C. But in the training dataset I have A dataset with 70% volume, B with 25% and C with 5%. Most of time my results are overfit to A. Can you please suggest how can I solve this problem?

I write long lists of techniques to try and think about the best ways to get past this problem. I finally took the advice of one of my students:



Perhaps one of your upcoming blog posts could address the problem of training a model to perform against highly imbalanced data, and outline some techniques and expectations.

Frustration!

Imbalanced data can cause you a lot of frustration.

You feel very frustrated when you discovered that your data has imbalanced classes and that all of the great results you thought you were getting turn out to be a lie.

The next wave of frustration hits when the books, articles and blog posts don't seem to give you good advice about handling the imbalance in your data.

Relax, there are many options and we're going to go through them all. It is possible, you can build predictive models for imbalanced data.

Want to Get Started With Imbalance Classification?

Take my free 7-day email crash course now (with sample code).

Click to sign-up and also get a free PDF Ebook version of the course.

[Download Your FREE Mini-Course](#)

What is Imbalanced Data?

Imbalanced data typically refers to a problem with classification problems where the classes are not represented equally.

For example, you may have a 2-class (binary) classification problem with 100 instances (rows). A total of 80 instances are labeled with Class-1 and the remaining 20 instances are labeled with Class-2.

This is an imbalanced dataset and the ratio of Class-1 to Class-2 instances is 80:20 or more concisely 4:1.

You can have a class imbalance problem on two-class classification problems as well as multi-class classification problems. Most techniques can be used on either.

The remaining discussions will assume a two-class classification problem because it is easier to think about and describe.

Imbalance is Common

Most classification data sets do not have exactly equal number of instances in each class, but a small difference often does not matter.

There are problems where a class imbalance is not just common, it is expected. For example, in datasets like those that characterize fraudulent transactions are imbalanced. The vast majority of the transactions will be in the “Not-Fraud” class and a very small minority will be in the “Fraud” class.

Another example is customer churn datasets, where the vast majority of customers stay with the service (the “No-Churn” class) and a small minority cancel their subscription (the “Churn” class).

When there is a modest class imbalance like 4:1 in the example above it can cause problems.

Accuracy Paradox

The [accuracy paradox](#) is the name for the exact situation in the introduction to this post.

It is the case where your accuracy measures tell the story that you have excellent accuracy (such as 90%), but the accuracy is only reflecting the underlying class distribution.

It is very common, because classification accuracy is often the first measure we use when evaluating models on our classification problems.

Put it All On Red!

What is going on in our models when we train on an imbalanced dataset?

As you might have guessed, the reason we get 90% accuracy on an imbalanced data (with 90% of the instances in Class-1) is because our models look at the data and cleverly decide that the best thing to do is to always predict “Class-1” and achieve high accuracy.

This is best seen when using a simple rule based algorithm. If you print out the rule in the final model you will see that it is very likely predicting one class regardless of the data it is asked to predict.

8 Tactics To Combat Imbalanced Training Data

We now understand what class imbalance is and why it provides misleading classification accuracy.

So what are our options?

1) Can You Collect More Data?

You might think it's silly, but collecting more data is almost always overlooked.

Can you collect more data? Take a second and think about whether you are able to gather more data on your problem.

A larger dataset might expose a different and perhaps more balanced perspective on the classes.

More examples of minor classes may be useful later when we look at resampling your dataset.

2) Try Changing Your Performance Metric

Accuracy is not the metric to use when working with an imbalanced dataset. We have seen that it is misleading.

There are metrics that have been designed to tell you a more truthful story when working with imbalanced classes.

I give more advice on selecting different performance measures in my post [“Classification Accuracy is Not Enough: More Performance Measures You Can Use”](#).

In that post I look at an imbalanced dataset that characterizes the recurrence of breast cancer in patients.

From that post, I recommend looking at the following performance measures that can give more insight into the accuracy of the model than traditional classification accuracy:

- **Confusion Matrix:** A breakdown of predictions into a table showing correct predictions (the diagonal) and the types of incorrect predictions made (what classes incorrect predictions were assigned).
- **Precision:** A measure of a classifiers exactness.
- **Recall:** A measure of a classifiers completeness
- **F1 Score (or F-score):** A weighted average of precision and recall.

I would also advice you to take a look at the following:

- **Kappa (or Cohen’s kappa):** Classification accuracy normalized by the imbalance of the classes in the data.
- **ROC Curves:** Like precision and recall, accuracy is divided into sensitivity and specificity and models can be chosen based on the balance thresholds of these values.

You can learn a lot more about using ROC Curves to compare classification accuracy in our post [“Assessing and Comparing Classifier Performance with ROC Curves”](#).

Still not sure? Start with kappa, it will give you a better idea of what is going on than classification accuracy.

3) Try Resampling Your Dataset

You can change the dataset that you use to build your predictive model to have more balanced data.

This change is called sampling your dataset and there are two main methods that you can use to even-up the classes:

1. You can add copies of instances from the under-represented class called over-sampling (or more formally sampling with replacement), or

2. You can delete instances from the over-represented class, called under-sampling.

These approaches are often very easy to implement and fast to run. They are an excellent starting point.

In fact, I would advise you to always try both approaches on all of your imbalanced datasets, just to see if it gives you a boost in your preferred accuracy measures.

You can learn a little more in the the Wikipedia article titled "[Oversampling and undersampling in data analysis](#)".

Some Rules of Thumb

- Consider testing under-sampling when you have an a lot data (tens- or hundreds of thousands of instances or more)
- Consider testing over-sampling when you don't have a lot of data (tens of thousands of records or less)
- Consider testing random and non-random (e.g. stratified) sampling schemes.
- Consider testing different resampled ratios (e.g. you don't have to target a 1:1 ratio in a binary classification problem, try other ratios)

4) Try Generate Synthetic Samples

A simple way to generate synthetic samples is to randomly sample the attributes from instances in the minority class.

You could sample them empirically within your dataset or you could use a method like Naive Bayes that can sample each attribute independently when run in reverse. You will have more and different data, but the non-linear relationships between the attributes may not be preserved.

There are systematic algorithms that you can use to generate synthetic samples. The most popular of such algorithms is called SMOTE or the Synthetic Minority Over-sampling Technique.

As its name suggests, SMOTE is an oversampling method. It works by creating synthetic samples from the minor class instead of creating copies. The algorithm selects two or more similar instances (using a distance measure) and perturbing an instance one attribute at a time by a random amount within the difference to the neighboring instances.

Learn more about SMOTE, see the original 2002 paper titled "[SMOTE: Synthetic Minority Over-sampling Technique](#)".

There are a number of implementations of the SMOTE algorithm, for example:

- In Python, take a look at the "[UnbalancedDataset](#)" module. It provides a number of implementations of SMOTE as well as various other resampling techniques that you could try.
- In R, the [DMwR package](#) provides an implementation of SMOTE.
- In Weka, you can use the [SMOTE supervised filter](#).

5) Try Different Algorithms

As always, I strongly advice you to not use your favorite algorithm on every problem. You should at least be spot-checking a variety of different types of algorithms on a given problem.

For more on spot-checking algorithms, see my post “Why you should be Spot-Checking Algorithms on your Machine Learning Problems”.

That being said, decision trees often perform well on imbalanced datasets. The splitting rules that look at the class variable used in the creation of the trees, can force both classes to be addressed.

If in doubt, try a few popular decision tree algorithms like C4.5, C5.0, CART, and Random Forest.

For some example R code using decision trees, see my post titled “[Non-Linear Classification in R with Decision Trees](#)”.

For an example of using CART in Python and scikit-learn, see my post titled “[Get Your Hands Dirty With Scikit-Learn Now](#)”.

6) Try Penalized Models

You can use the same algorithms but give them a different perspective on the problem.

Penalized classification imposes an additional cost on the model for making classification mistakes on the minority class during training. These penalties can bias the model to pay more attention to the minority class.

Often the handling of class penalties or weights are specialized to the learning algorithm. There are penalized versions of algorithms such as penalized-SVM and penalized-LDA.

It is also possible to have generic frameworks for penalized models. For example, Weka has a [CostSensitiveClassifier](#) that can wrap any classifier and apply a custom penalty matrix for miss classification.

Using penalization is desirable if you are locked into a specific algorithm and are unable to resample or you're getting poor results. It provides yet another way to “balance” the classes. Setting up the penalty matrix can be complex. You will very likely have to try a variety of penalty schemes and see what works best for your problem.

7) Try a Different Perspective

There are fields of study dedicated to imbalanced datasets. They have their own algorithms, measures and terminology.

Taking a look and thinking about your problem from these perspectives can sometimes shake loose some ideas.

Two you might like to consider are **anomaly detection** and **change detection**.

[Anomaly detection](#) is the detection of rare events. This might be a machine malfunction indicated through its vibrations or a malicious activity by a program indicated by its sequence of system calls. The events are rare and when compared to normal operation.

This shift in thinking considers the minor class as the outliers class which might help you think of new ways to separate and classify samples.

[Change detection](#) is similar to anomaly detection except rather than looking for an anomaly it is looking for a change or difference. This might be a change in behavior of a user as observed by usage patterns or bank transactions.

Both of these shifts take a more real-time stance to the classification problem that might give you some new ways of thinking about your problem and maybe some more techniques to try.

8) Try Getting Creative

Really climb inside your problem and think about how to break it down into smaller problems that are more tractable.

For inspiration, take a look at the very creative answers on Quora in response to the question “[In classification, how do you handle an unbalanced training set?](#)”

For example:



Decompose your larger class into smaller number of other classes...

...use a One Class Classifier... (e.g. treat like outlier detection)

...resampling the unbalanced training set into not one balanced set, but several. Running an ensemble of classifiers on these sets could produce a much better result than one classifier alone

These are just a few of some interesting and creative ideas you could try.

For more ideas, check out these comments on the reddit post “[Classification when 80% of my training set is of one class](#)”.

Pick a Method and Take Action

You do not need to be an algorithm wizard or a statistician to build accurate and reliable models from imbalanced datasets.

We have covered a number of techniques that you can use to model an imbalanced dataset.

Hopefully there are one or two that you can take off the shelf and apply immediately, for example changing your accuracy metric and resampling your dataset. Both are fast and will have an impact straight away.

Which method are you going to try?

A Final Word, Start Small

Remember that we cannot know which approach is going to best serve you and the dataset you are working on.

You can use some expert heuristics to pick this method or that, but in the end, the best advice I can give you is to “become the scientist” and empirically test each method and select the one that gives you the best results.

Start small and build upon what you learn.

Want More? Further Reading...

There are resources on class imbalance if you know where to look, but they are few and far between.

I’ve looked and the following are what I think are the cream of the crop. If you’d like to dive deeper into some of the academic literature on dealing with class imbalance, check out some of the links below.

Books

- [Imbalanced Learning: Foundations, Algorithms, and Applications](#), 2013.
- [Learning from Imbalanced Data Sets](#), 2018.

Papers

- [Data Mining for Imbalanced Datasets: An Overview](#)
- [Learning from Imbalanced Data](#)
- [Addressing the Curse of Imbalanced Training Sets: One-Sided Selection](#) (PDF)
- [A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data](#)

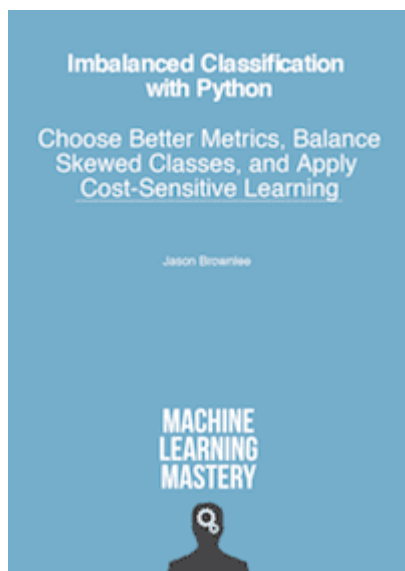
Did you find this post useful? Still have questions?

Leave a comment and let me know about your problem and any questions you still have about handling imbalanced classes.

Get a Handle on Imbalanced Classification!

Develop Imbalanced Learning Models in Minutes

...with just a few lines of python code



Discover how in my new Ebook:
[Imbalanced Classification with Python](#)

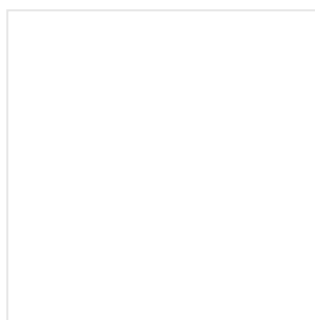
It provides **self-study tutorials** and **end-to-end projects** on:
Performance Metrics, Undersampling Methods, SMOTE, Threshold Moving, Probability Calibration, Cost-Sensitive Algorithms
and much more...

Bring Imbalanced Classification Methods to Your Machine Learning Projects

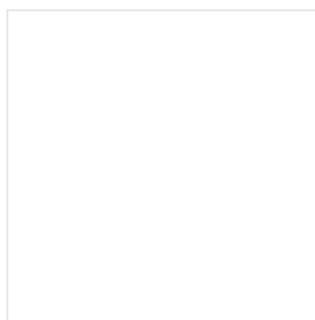
SEE WHAT'S INSIDE



More On This Topic



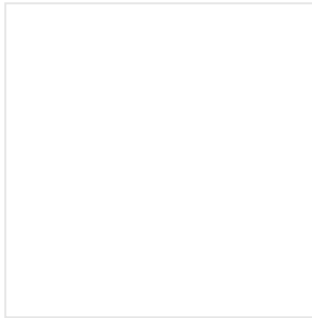
How to Define Your Machine Learning Problem



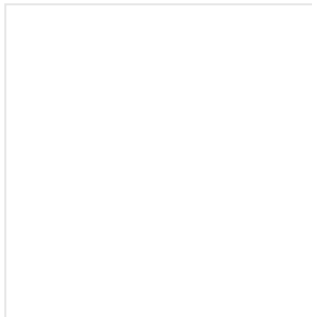
Why you should be Spot-Checking Algorithms on your...



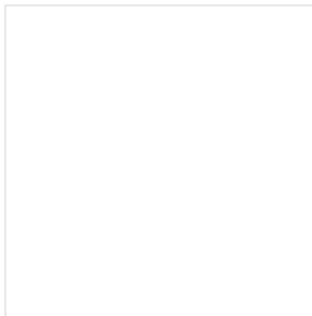
Quick and Dirty Data Analysis for your Machine...



How to Layout and Manage Your Machine Learning Project



What Is Holding You Back From Your Machine Learning Goals?



Get Your Dream Job in Machine Learning by Delivering Results

About Jason Brownlee

Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.

[View all posts by Jason Brownlee →](#)

306 Responses to *8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset*

Sebastian Raschka August 26, 2015 at 2:47 am <#>

REPLY 

Interesting survey! Maybe it would be worthwhile to mention semi-supervised techniques to utilize unlabeled data? There are many different approaches, if you are interested, check out this nice survey: X. Zhu, "Semi-Supervised Learning Literature Survey," Technical Report 1530, Univ. of Wisconsin-Madison, 2006.

Transfer learning can also be interesting in context of class imbalances for using unlabeled target data as regularization term to learn a discriminative subspace that can generalize to the target domain: Si S, Tao D, Geng B. Bregman divergence-based regularization for transfer subspace learning. IEEE Trans on Knowledge and Data Engineering 2010;22:929–42.

Or for the very extreme cases 1-class SVM 🤪 Scholkopf B, Platt JC, Shawe-Taylor JC, Smola AJ, Williamson RC. Estimating the support of a high-dimensional distribution. Neural Computation 2001;13:1443–71.

Igor Vieira October 29, 2015 at 8:26 am <#>

REPLY 

Great post!!

Jason Brownlee November 16, 2015 at 8:14 pm <#>

REPLY 

Thanks Igor

Foram January 9, 2020 at 6:07 am <#>

REPLY 

I have one question like can we sampled testing data as well? or just training? Because I have unbalanced data and when i use sampling methods on training data and predict on unsampled testing set it gives me worst output.

Thanks.

Jason Brownlee January 9, 2020 at 7:33 am <#>

REPLY 

Only training data.

priyanka rana September 6, 2021 at 10:39 pm <#>

Thank you so much for the post. thats a great help.

I am working on highly imbalanced dataset, where minority class has 15 samples while the majority one has 9000 samples.

I am trying various approaches for oversampling to train ResNet deep learning model for the classification of classes.

Considering 20% of data for validation and another 20% for testing, leaves only 2 images in test set and 3 for validation set for minority class.

Due to this I am getting very fluctuating results each time i train the model with same hyper parameters. Because in one training session both images are identified while in another one none or only 1 image is identified, which hugely impacts the F1 score.

I want to have reproducible results , but at the same time do not want to augment test set images.

Can you please give me a suggestion on this.

Many thanks

Adrian Tam September 7, 2021 at 6:18 am <#>

I think you already reached the limit of the data you have. I would really try to do augmentation, at least after the training set is created.

Jingchen November 16, 2015 at 11:26 am <#>

REPLY 

Hi Jason, this is a very helpful post. Save me a lot of time for checking detailed solutions and it's eye-opening.

Thanks

Jason Brownlee November 16, 2015 at 8:14 pm <#>

REPLY 

Glad to hear it Jingchen!

Haifeng Liu November 21, 2015 at 1:54 pm <#>

REPLY 

This is really a great and helpful post!

Jason Brownlee July 8, 2016 at 7:06 am <#>

REPLY 

Glad to hear it Haifeng.

Parinya.hi January 13, 2016 at 3:47 pm <#>

REPLY 

Hi Jason, really cool and helpful post. I have done some but for my case seems quite difficult because of most of predictor values are flag (0 or 1).

Abraham B.Gabriel January 18, 2016 at 1:07 pm <#>

REPLY 

Hi Jason. You just saved a life. (Quite literally).Thanks a lot for the article.

Jason Brownlee July 8, 2016 at 7:06 am <#>

REPLY 

Wow, well best of luck Abraham.

Kazim K February 1, 2018 at 2:59 am <#>

REPLY 

Great article. Thanks for your effort.

Jason Brownlee February 1, 2018 at 7:24 am <#>

REPLY 

Thanks.

Vered Shwartz February 1, 2016 at 7:52 am <#>

REPLY 

Thanks for a very helpful post! I have an unbalanced dataset (50:1 negative-positive ratio) and I've tried some of the techniques you discussed. I was wondering about subsampling – if I train my model over all the positive instances and an equal number of negative instances, there is a lot of

unused data. Is there a reasonable way that I can perform several iterations, each with different negative instances, and combine the results into one model? Thanks in advance.

Jon D. August 19, 2017 at 10:44 am <#>

REPLY 

The situation you describe is exactly what oversampling is. Just use fewer iterations than what you would with undersampling.

Swap July 16, 2019 at 1:13 am <#>

REPLY 

Hi Jason,
Brilliant summary I have tried most of it and my predictions still are not correct. Wondering if you can nudge me in the right direction.

Background

I am doing a 10 class classification. One class is the dominant one making 30% of the sample. My predictions still classifies most as 30%.

Things I have tried

Smote,

One vs all

Different algorithms

Using F1 score

Jason Brownlee July 16, 2019 at 8:20 am <#>

REPLY 

Perhaps some feature engineering?

More general ideas here:

<http://machinelearningmastery.com/machine-learning-performance-improvement-cheat-sheet/>

bikiltu guteta June 26, 2020 at 4:16 am <#>

REPLY 

dear Jason. do you have any tutorial on conditional random fields for text preparation?

Jason Brownlee June 26, 2020 at 5:40 am <#>

Not at this stage, perhaps I will write about them in the future.

Jason Brownlee February 3, 2016 at 8:50 pm #

REPLY ↩

Great and relevant post: [Dealing with imbalanced data: undersampling, oversampling and proper cross-validation](#) , by Marco Altini.

bassel May 16, 2016 at 4:19 am #

REPLY ↩

is there a way in “sklearn” module to Penalize the machine learning algorithm and make a penalize model
by adding an additional cost on the model for making classification mistakes on the minority class during training, or i must implement the algorithm from scratch

Jason Brownlee July 8, 2016 at 7:08 am #

REPLY ↩

I believe you can use sample weights.

Google found this on StackOverflow:

<http://stackoverflow.com/questions/20082674/unbalanced-classification-using-randomforestclassifier-in-sklearn>

bikiltu guteta June 26, 2020 at 4:20 am #

REPLY ↩

thank you for your best tutorial on cleaning text in machine learning, but i have a question on that how can tokenize large file or the whole documents in my dataset at once doing?

Jason Brownlee June 26, 2020 at 5:41 am #

REPLY ↩

You may need to use progressive loading or a python generator to load and process the input in batches.

Alternately, you could use a machine with very large RAM and load all data into ram for processing.

The former is more common.

Adeyemi February 4, 2016 at 8:43 am #

REPLY ↩

Great. I want to try the SMOTE with Weka, is there any simple sample tutorial to use the SMOTE supervised filter? I need guidance

Jason Brownlee July 8, 2016 at 7:09 am #

REPLY ↩

I don't have a tutorial, but these notes on stack overflow might help:
<http://stackoverflow.com/questions/22632932/how-to-set-parameters-in-weka-to-balance-data-with-smote-filter>

Matthew R. Versaggi April 11, 2016 at 9:29 pm #

REPLY ↩

Fantastic, just what the doctor ordered !

Thank you for your efforts, it's enabling the advancing of the field ...

Jason Brownlee July 8, 2016 at 7:09 am #

REPLY ↩

You're welcome Matthew.

Ella April 23, 2016 at 5:52 am #

REPLY ↩

Thank you, very helpful post.

I am using "UnbalancedDataset" module in Python to perform over sampling with synthetic data generation (SMOTE/Random) I am wondering if there is any smart way to find the best ratio for over sampling ?

I have a binary classification problem with imbalanced data with the rate of 1/5.

Jason Brownlee July 8, 2016 at 7:11 am #

REPLY ↩

Generally, I would advise systematic experimentation to discover good or best configuration for your problem.

Bar Geron June 4, 2016 at 10:26 pm #

REPLY ↩

Hi,

Great article!

it will be much appreciated if you can help with the following question:

I've used the over sampling approach and change the ratio of my binary target value from 1:10 to 1:1.
the problem is that i still don't how to check the model performance on the ratio of 1:10.

how do i know what will be the gap of impact between the real world and the 1:1 ratio ?

Jason Brownlee June 14, 2016 at 8:26 am #

REPLY ↩

A good idea would be to hold back a validation dataset, say split the dataset in half.

Try various rebalancing methods and modeling algorithms with cross validation, then use the held back dataset to confirm any findings translate to a sample of what the actual data will look like in practice.

Hua Yang July 8, 2016 at 5:03 am #

REPLY ↩

Hi Jason,

I have the same question as Bar Geron.

what did you mean by saying “then use the held back dataset to confirm any findings translate to a sample of what the actual data will look like in practice”?

Could you please explain your it with more details?

Thank you!

Jason Brownlee July 8, 2016 at 7:13 am #

REPLY ↩

I meant that you can use cross validation on the rebalanced dataset to estimate the performance of models on unseen data.

You can then build a final model and evaluate it's performance on the held out dataset.

This will allow you to see whether findings from resampling during cross validation translate over to “unseen” data.

David F July 1, 2016 at 1:26 am #

REPLY ↩

Pretty useful article. Thank you very much!

Jason Brownlee July 1, 2016 at 5:41 am #

REPLY ↩

You're welcome David.

Kaustubh Patil July 3, 2016 at 4:03 am #

REPLY ↩

Another tactic is to change the decision threshold on the posterior probability. We have shown that this particularly works well with bagging ensemble which are known to give good posterior estimates. See "Reviving Threshold-Moving: a Simple Plug-in Bagging Ensemble for Binary and Multiclass Imbalanced Data" <http://arxiv.org/abs/1606.08698>.

Disclaimer: I am the last author of this paper

Jason Brownlee July 3, 2016 at 7:16 am #

REPLY ↩

Great, thanks for the pointer Kaustubh.

Manish July 11, 2016 at 2:40 am #

REPLY ↩

Great article! Very helpful.

Jason Brownlee July 11, 2016 at 5:40 am #

REPLY ↩

Thanks Manish.

ankita July 22, 2016 at 10:42 am #

REPLY ↩

sir,, in future which issues related to classification problem which can be solved?

RCB August 3, 2016 at 1:13 am #

REPLY ↩

I consider this a non-issue. There's no statistical method or machine learning algorithm I know of that requires balanced data classes. Furthermore, if *reality is unbalanced*, then you want your algorithm to learn that!

Consider the problem of trying to predict two outcomes, one of which is much more common than the other. Suppose there is a region in feature space in which the two classes very strongly overlap. Then the prediction in this region will depend on the frequency of each class that fall in this region in the training set. If you've "balanced" the data by hugely biasing it toward the rare class, then your model will predict something like 50% probability of each, when the truth is probably very different.

The problem, IMO, isn't unbalance. The world is unbalanced. The problem is that rare classes are poorly represented unless the datasets are quite large. In other words, it's a sample size problem. A lot of the

difficulty can be cleared up (as the author points out) but looking at false positive and false negative rates, not just generic “accuracy”.

Jason Brownlee August 3, 2016 at 8:17 am #

REPLY ↩

Thought provoking perspective RCB, thanks for sharing.

I have to disagree that this is a non-issue in practice. At least for me, I almost always seem to get better results when I “handle” the class imbalance.

As a test, grab an unbalanced dataset from the UCI ML repo and do some small experiments. I think you will quickly realize that either you need to change up your performance measure and start exploring resampling or instance weighting methods to get any kind of traction on the problem.

In fact, this might make a nice blog post tutorial, stay tuned.

RCB August 5, 2016 at 8:20 am #

REPLY ↩

At the end of the day, performance is what matters, so I won't be so foolish as to take a hard-line stance. But I'm inclined to think that there is always a better alternative to “rebalancing” the data, i.e. one should never have to do it, in theory.

Your model is doing its best to minimize the loss function you specify. If this is just classification accuracy, then it's quite plausible that the best classifier is one that always picks the vastly-more-common class. What this is telling you is that the model has not seen enough examples of the rare class to be able to distinguish them from the common class. Failing that, it simply says “forget it: just always predict the most common class!” If you're only interested in 1-0 classification accuracy, then that is the best model, period, given the loss function and dataset you provided.

Now, if you find yourself thinking that this is a very unsatisfactory outcome, ask yourself why! Probably it's because misclassification of the rare class is a lot worse than the alternative. i.e., false negatives are a lot worse than false positives. Perhaps you are diagnosing cancers, or catching failed products. Well, clearly this problem is solved by choosing a more appropriate loss function – not biasing the data! Just make the “cost” of a false negative much greater than the cost of a false positive. This will give you a cost function that better represents your priorities, while still maintaining a realistic dataset. Rebalancing does neither!

Also: By hugely rebalancing (read: hugely biasing) the model, you are training on a dataset that will be vastly different from the dataset it will actually be tested on, once implemented. That strikes me as a bad thing. Train it for the real world.

IMO.

Jason Brownlee August 5, 2016 at 8:25 am #

REPLY ↩

A very reasoned comment, thanks RCB.

Usman June 17, 2017 at 11:39 am #

will converting imbalance dataset to balanced dataset, (by decreasing the number of normal class instances) increases the false positives? In a case of cancer detection, we might end up predicting more cancer patients while there were not. Is my assumption wrong?

Jason Brownlee June 18, 2017 at 6:29 am #

It may, you must balance transforms of your dataset with the goal of the project. Choose your model evaluation metrics carefully.

DavidFarago January 12, 2021 at 6:57 am #

I am reading a lot about rebalancing of imbalanced data and was wondering whether it is a good idea to have a completely different distribution for your train set and test set. Looking for an answer, I found this blog post, which sounds like rebalancing is a reasonable thing to do.

Then I read RCB's comments and now I am wondering again.

@Jason: Is your experience 4 years later still that rebalancing leads to better results? Should you use a train-dev set (a set between training set and dev set), so that you can measure a data mismatch error between train-dev set and dev set.

Jason Brownlee January 12, 2021 at 7:58 am #

It is invalid to change the distribution of the test set.

It is a great idea to change the distribution of your training set to balance or even overemphasise a minority class.

Balancing is one method that works sometimes.

There are many techniques you can use, I even wrote a book on the topic, you can start here:

<https://machinelearningmastery.com/start-here/#imbalanced>

Chris John August 4, 2016 at 8:04 pm #

REPLY ↩

Thanks for this!

Jason Brownlee August 5, 2016 at 5:27 am #

REPLY ↩

You're welcome Chris.

Shravan Kumar Parunandula July 28, 2019 at 5:57 pm #

REPLY ↩

What do you suggest on using conditional gans in generating synthetic samples, as in tactic 3.

Jason Brownlee July 29, 2019 at 6:11 am #

REPLY ↩

It would be tactic 4.

Try it and see.

Simon August 5, 2016 at 9:32 pm #

REPLY ↩

Hi Jason, can windowing a long time series be used as a sampling technique? That way, we get many smaller segments of the same time series, and if we label them up the same, we can consider them as larger data to extract features from, can we not? In that case, what criteria should we look at? Long range correlations? I know that the statistics can change, so usually a non-stationary time series can be changed to a stationary time series either through filtering or some sort of background levelling (to level the trend).

Am I thinking in the right directions?

The second part of my question is, if we do not go for sampling methods and consider the whole time series as one data point, what classification and feature extraction algorithm should I look for?

Eagerly waiting for your reply. Many thanks
Simon

Evelyn August 11, 2016 at 7:25 am #

REPLY ↩

Hi Jason,

I have a question about how should we deal with the over sampled dataset. There are two ways come to my mind and I am now going with the first one, which seem very overfitting.

- 1- Oversample the whole dataset, then split it to training and testing sets (or cross validation).
- 2- After splitting the original dataset, perform oversampling on the training set only and test on the original data test set.

My dataset has multi-labels, there is one majority label and number of samples for others label are quite small, some even has the ratio 100:1, how should I deal with this dataset?

Thanks a lot!

Jason Brownlee August 15, 2016 at 11:25 am #

REPLY ↩

I would suggest separating out a validation dataset for later use.

I would suggest applying your procedure (say oversampling) within the folds of a cross validation process with possible. Otherwise just on the training dataset for a train/test split.

Amr August 28, 2017 at 1:47 am #

REPLY ↩

but then how would the accuracy measure on the test dataset (with the imbalanced classes) be relevant?

Jason Brownlee August 28, 2017 at 6:50 am #

REPLY ↩

You would evaluate the model on the test set directly, with no change to the test set.

Marta AZ August 11, 2016 at 11:07 pm #

REPLY ↩

Thank you for the article. It is a very good and effective summary of a wide a complicated problem. Please go on with your blog!

Jason Brownlee August 15, 2016 at 11:24 am #

REPLY ↩

Thanks Marta

Mohammed Tantawy August 30, 2016 at 11:37 am #

REPLY ↩

Great Post Jason

Jason Brownlee August 31, 2016 at 8:44 am #

REPLY ↩

Thanks Mohammed, I'm glad you found it useful.

Linara September 5, 2016 at 8:01 pm #

REPLY ↩

Thanks for tips!

Can you please elaborate more or give some useful sources for the Penalized models? I am using logistic regression with standard log likelihood loss function ($-\text{mean}(\text{teacher} * \log(\text{predicted}) + (1 - \text{teacher}) * \log(1 - \text{predicted}))$) and I want to know what exactly is a correct way to make it pay more attention to 1-class, because my data has about 0.33% of 1-class examples and all the others are 0-class.

Jason Brownlee September 6, 2016 at 9:48 am #

REPLY ↩

Perhaps you could experiment with weighting observations for one class or another. I have seen this be very effective with regression methods.

Linara September 6, 2016 at 6:30 pm #

REPLY ↩

The main question is more about what part should be more “important”? I have to put more weight to the error part that is obtained from the rare class (e.g. to have something like $-\text{mean}(0.9 * \text{teacher} * \log(\text{predicted}) + 0.1 * (1 - \text{teacher}) * \log(1 - \text{predicted}))$) or other way around – I have to “penalize” big class (e.g. $-\text{mean}(0.1 * \text{teacher} * \log(\text{predicted}) + 0.9 * (1 - \text{teacher}) * \log(1 - \text{predicted}))$)? Because penalizing is more about that I have to do something with big class and weighting is the thing that has to be larger for the rare class and this terminology completely confuses me. And does the weights have something to do by value with the frequency of the rare class?

Jason Brownlee September 7, 2016 at 10:26 am #

REPLY ↩

Yes, the weight is to rebalance the observations for each class, so the sum of observations for each class are equal.

K Rajesh September 11, 2016 at 2:53 am #

Sir, I am also working on such type of imbalanced multi-class problem. In my case, accuracy values are over dependent on normalization procedure.

Following discussion will give an overview of my problem.

Is it possible to do feature normalization with respect to class. EX: 10×10 data matrix with two class. Each class of size 5×5. Now normalize 25 features of class 1 and 25 features of class 2 separately. Is this process acceptable.

Jason Brownlee September 12, 2016 at 8:28 am #

No. You will not know the class of new data in the future, therefore you won't know what procedure to use.

Abdul October 20, 2016 at 4:59 pm #

Sir, How to specify weights in RF.

I need help in specifying weights for each split instead of gini indexing.

Your response will be appreciated.

Jason Brownlee October 21, 2016 at 8:33 am #

I don't know about weighted random forest or weighted gini calculation, sorry Abdul.

Erik Yao September 23, 2016 at 5:11 am #

REPLY ↩

Thank you, Jason! I am playing around with a 19:1 data set and your post provides a lot of techniques to handle the imbalance. I am very interested to try them one by one to see what can I get at best.

Jason Brownlee September 23, 2016 at 8:30 am #

REPLY ↩

I'm glad to hear it, thanks Erik.

Gilmar October 4, 2016 at 1:30 pm #

REPLY ↩

Great post Jason!!!

Jason Brownlee October 4, 2016 at 2:56 pm #

REPLY ↩

Thanks Gilmar, I'm glad you found it useful.

charisfauzan October 13, 2016 at 3:47 pm #

REPLY ↩

Great post sir, It is useful for me...

Jason Brownlee October 14, 2016 at 8:58 am #

REPLY ↩

I'm very glad to hear that charisfauzan.

Sarah November 16, 2016 at 6:27 am #

REPLY ↩

Great post. Read it almost 5 times.

Jason Brownlee November 16, 2016 at 9:34 am #

REPLY ↩

Thanks Sarah, I'm glad to hear it.

Chris January 3, 2017 at 9:58 pm #

REPLY ↩

Check my question here please. I don't know what happens.

<http://cs.stackexchange.com/questions/68212/big-number-of-false-positives-in-binary-classification>

Jason Brownlee January 4, 2017 at 8:52 am #

REPLY ↩

Hi Chris, perhaps you could write a one sentence summary of your problem?

Licheng January 15, 2017 at 7:25 am #

REPLY ↩

Hi Jason, Thanks for the great article.

I have a question about imbalanced multiclass problem (7 classes). I tried oversampling SMOTE and it seems what it does is to match the class with least samples to the class with most samples, nothing changes with the other classes. I wonder if this is how it should be.

Jason Brownlee January 16, 2017 at 10:34 am #

REPLY ↩

Sorry Licheng, I don't have any tutorials for SMOTE at the moment. It is an area I plan to cover soon.

Are you using R or Python?

You can learn more about the SMOTE method here:

<https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume16/chawla02a-html/chawla2002.html>

KT March 2, 2017 at 8:49 am #

REPLY ↩

SMOTE doesnt seem to handle multiclass problems

Santosh September 25, 2018 at 4:08 pm #

REPLY ↩

Please try ADASYN...It works well in case of multiclass prediction

Natheer Alabsi January 25, 2017 at 3:52 pm #

REPLY ↩

Hi Jason,

Is it acceptable to intentionally choose an imbalanced subset of the two classes available in the data for training if that will increase the accuracy of the model.

Jason Brownlee January 26, 2017 at 4:43 am #

REPLY ↩

I like your thinking Natheer, try and see!

Consider an ensemble of a suite of models, biased in different ways.

Do not use the accuracy measure to evaluate your results though, it will not give a clear idea of how well your model is actually performing.

Natheer Alabsi January 26, 2017 at 9:14 pm #

REPLY ↩

Thanks for the reply. But I want to use only one sample from the negative class(not buy the product) and a large sample from the positive class(buy the product). I noticed it improved the accuracy so much.

Jason Brownlee January 27, 2017 at 12:06 pm #

REPLY ↩

Hi Natheer, in this case accuracy would be an invalid measure of performance.

Natheer Alabsi January 27, 2017 at 12:12 pm #

I know accuracy of the overall model is meaningless but it is best increase in recall over other situations.

Jingjing February 27, 2017 at 1:31 pm #

REPLY ↩

Thank you for your effort. It's really helpful! Do you have some imbalanced data sets? I can not find some perfect data sets for my algorithm.

Jason Brownlee February 28, 2017 at 8:10 am #

REPLY ↩

Consider searching on this site:
<http://archive.ics.uci.edu/ml/>

deep March 22, 2017 at 7:02 pm #

REPLY ↩

Hi Jason,

Thanks for uploading a very nice informative article about imbalance dataset handling. I am trying to build deep learning model for classification. I have data set consist of approx 100k samples with around 36k features and six different classes with imbalanced class distribution. The largest class has approx 48k samples while smallest one has around 2k samples. Other classes have sample numbers like 18k,15k, 12k and 5k. I am considering the usage of smote for synthetic data generation for all small classes(18k-2k) up to 48K (biggest class). Is that scientifically appropriate approach? If not what else I can do?

Jason Brownlee March 23, 2017 at 8:48 am #

REPLY ↩

Try it and see.

wanida saetang April 3, 2017 at 2:27 pm #

REPLY ↩

Hi Jason, Thank for the great article. I'm interested in this. It was very helpful to me.

Jason Brownlee April 4, 2017 at 9:12 am #

REPLY ↩

You're welcome wanida.

Afifatul Mukaroh April 6, 2017 at 10:53 am #

REPLY ↩

"become the scientist" and empirically test each method and select the one that gives you the best results.

Oh my God. I like your advice.

Thanks also for your writing. It's very insightful, especially for me who's facing imbalance dataset right now.

Thanks a lot.

Jason Brownlee April 9, 2017 at 2:37 pm #

REPLY ↩

I'm glad to hear it.

Abtohi April 20, 2017 at 1:59 am #

REPLY ↩

nice post

Jason Brownlee April 20, 2017 at 9:30 am #

REPLY ↩

I'm glad you found it useful.

Camila April 28, 2017 at 5:59 am <#>

REPLY 

Thanks very much! I'm Chilean and I was looking for information in spanish but I don't discovered nothing. But I tried in english and you were very helpful!

Jason Brownlee April 28, 2017 at 7:57 am <#>

REPLY 

Thanks Camila, I'm glad to hear that.

Mazid OSSENI May 8, 2017 at 11:30 pm <#>

REPLY 

Thanks you for this post! It's very insightful, and helpful for an overall introduction to imbalanced problems. I've still had a couple of question, if you mind.

1) When using Penalized Models, how do we analyse the performance of the classifiers? Should we just used the classic metrics (precision, accuracy, f1_score, ...) or we must used a weighted metrics? (i don't know if you are familiar with it but in sklearn there's an optional sample_weights parameter in the metrics calculation)

2) Also what do you suggest as type of weights for the penalized Models?

PS: I read you're article on the metrics too, but i didn't find my answer there.

Thx

Jason Brownlee May 9, 2017 at 7:44 am <#>

REPLY 

Yes, focus on the output performance regardless of the specific schemes used internally by the model.

I would recommend reading up on weighting schemes, but starting with a weighting that counteracts the base rates back to even would be a good start.

Riishikesh May 17, 2017 at 1:21 am <#>

REPLY 

Hi Jason,

Thanks a lot for the informative post.I want to try out a few of these tactics, but unable to find data sets with class imbalance. Do you have links to any data such data sets?

Jason Brownlee May 17, 2017 at 8:38 am <#>

REPLY 

There are many on the UCI Machine Learning Repository:

<http://archive.ics.uci.edu/ml/>

Kamal May 29, 2017 at 4:03 pm #

REPLY ↩

Hi Jason,

Thanks for your post. I just have a simple question. Lets say we have a dataset of 500 binary entries. And we're using logistic regression to find the best parameters fit. Also, assume we have only 5 ones and the rest are zeros. If we are to implement SMOTE, should we implement it to both the training and test sets or only to the training set?

Thanks,

Jason Brownlee June 2, 2017 at 12:21 pm #

REPLY ↩

Hood question Kamal,

SMOTE is only applied to the training dataset.

Ruby Chang June 9, 2017 at 8:43 pm #

REPLY ↩

This is great, Jason. A very informative overview on imbalanced data and greatly useful for the problem I have in hand where the minority class had less than 0.01% of the overall observations. I notice that this was posted almost two years ago, and do you know if there is new development on handling such data? Many thanks!

Jason Brownlee June 10, 2017 at 8:22 am #

REPLY ↩

Ouch, that is quite an imbalance.

These methods will be a great start. Once exhausted, I'm sure there is a suite of "state of the art" methods you could explore.

John J June 27, 2017 at 12:00 pm #

REPLY ↩

Google white papers for machine learning and rare diseases and you'll see some approaches for dealing with severe imbalance issues.

Jason Brownlee June 28, 2017 at 6:17 am <#>

REPLY 


Great suggestion John. Also intrusion detection problems and fraud problems.

vinod singh June 15, 2017 at 8:24 pm <#>

REPLY 

Hi Jason, I have dataset of 251 positive samples and 502 negative samples. I have just replicated my positive datasets to make it to 502 i.e., 100 percent replication of positive samples. This approach have significantly improved my results. But, I am confused whether my approach is correct or not. If it is correct, then is there any article of good journal to support my approach.

Jason Brownlee June 16, 2017 at 7:53 am <#>

REPLY 

Importantly, there is no “correct” approach in applied machine learning (or in life!).
Use the ideas here and elsewhere to design experiments in order to discover what works best on your specific problem.

Mark June 16, 2017 at 2:44 am <#>

REPLY 

Thank you this was very helpful. I appreciate your blog, keep it up!

Jason Brownlee June 16, 2017 at 8:04 am <#>

REPLY 

I'm glad to hear that, thanks Mark.

Thibaud June 17, 2017 at 12:19 am <#>

REPLY 

Thank you Jason for all the work you do. Do you think, it is possible to deal with unbalanced dataset by playing with decision threshold?

I mean, if you have a dataset with class 0 = 80% of observations and class 1 = 20% of observations, how about finding the optimal threshold by taking the one which separates the top 20% probabilities predictions from the lowest 80% probabilities predictions?

Thank you!

Jason Brownlee June 17, 2017 at 7:31 am <#>

REPLY 

Yes, especially on binary forecasting problems (e.g. ROC curves). Try it on your problem and see.

AJ July 17, 2017 at 11:39 am <#>

REPLY 

Hello Jason,

Thanks so much for all you do. You have helped me immensely!

I'm testing the difference between cost-sensitive learning and resampling for an imbalanced data set that has a small number of attributes to work with.

Would it ever be considered an acceptable practice to reverse/inverse the imbalance in a data set? If the priority was to predict an anomaly, and you were willing predict it at the expense of the majority, does this sound like a legitimate approach?

Thanks!

AJ

Jason Brownlee July 18, 2017 at 8:39 am <#>

REPLY 

Sure, try it. "Legitimate" is really defined by the model skill you can achieve.

Zahra July 19, 2017 at 12:07 pm <#>

REPLY 

I have the imbalanced multiclass classification problem with ratio 4:4:92. I want to ask if these techniques can work for my problem too..?? I means there is a huge difference like 4 & 4 to 92


Jason Brownlee July 19, 2017 at 4:09 pm <#>

REPLY 

Ouch, a challenging ratio.

I would recommend trying some of the resampling methods and selecting a metrics (e.g. log loss or similar) that best captures the goal of your project.

Ella August 17, 2017 at 12:14 am <#>

REPLY 

Hi Jason,

You mentioned that “decision trees often perform well on imbalanced datasets”. Does it mean that the imbalance data problem is not a big concern if decision tree method is employed?

Jason Brownlee August 17, 2017 at 6:43 am #

REPLY ↩

That can be the case, try and see on your data.

MU August 24, 2017 at 11:39 pm #

REPLY ↩

Hi Jason

I have a confusing situation with comparison of the test accuracies of different algorithms. I want to ask you if this is a true comparison if i make a synthetic imbalanced dataset from a real world dataset according to procedure A (ex:random %5 class noise). Apply my classification technique to that dataset than compare the test accuracies with an academic article X which is used another algorithm who made their own synthetic imbalanced dataset from the same realworld dataset according to procedure A. Is this a logic comparison according to you? Think that article and I use same real world dataset and same procedure to make that dataset imbalanced.

Thanks a lot from now

Jason Brownlee August 25, 2017 at 6:43 am #

REPLY ↩

Yes, if both tests started with the same source data.

Taniya September 15, 2017 at 2:17 pm #

REPLY ↩

Hi Jason.

can you help explain me about ADASYN method? I am confused about what is meant by minority data which is hard to learn.

Jason Brownlee September 16, 2017 at 8:37 am #

REPLY ↩

Sorry, I've not heard of it.

Try this search on google scholar:

<https://scholar.google.com/scholar?q=ADASYN+method>

Mahdi October 8, 2017 at 7:43 am #

REPLY ↩

Hello Jason,

Thanks for your valuable materials. I have an imbalanced data which all of its features are categorical and also response variable is ordinal. As I know SMOTE is only for continuous data .Is there any version of SMOTE for categorical and when I balance my dataset, are we supposed to consider the order in response variable? Thanks

Jason Brownlee October 8, 2017 at 8:43 am #

REPLY ↩

There may be, I would recommend searching on google scholar.

You could also try converting features to numeric and see if SMOTE and similar algorithms yield interesting results.

ips October 11, 2017 at 1:29 am #

REPLY ↩

Hi Jason

Thank you for your useful information.

in my journal about imbalanced class stated : “where more synthetic data is generated for minority class examples that are harder to learn”. what is the meaning of “harder to learn” ?

Thankyouu

Jason Brownlee October 11, 2017 at 7:55 am #

REPLY ↩

Can you post a link to the paper you are referencing?

ips October 11, 2017 at 12:22 pm #

REPLY ↩

the link : <http://www.ele.uri.edu/faculty/he/PDFfiles/adasyn.pdf>

Thankyou in advance

Jason Brownlee October 11, 2017 at 4:40 pm #

REPLY ↩

Thanks for sharing.

Generally, if you have questions about a paper, please contact the author of the paper.

Kuber October 19, 2017 at 8:57 am #

REPLY ↩

Hi Jason,

My dataset has 25:75 distribution of Churn: Not Churn. Should I consider this as imbalanced problem ? I am trying to use random forest on actual dataset to determine important features and then use logistic model without handling imbalanced classification problem. I am more familiar in python, and I am not sure if there is a verified oversampling algorithms currently that exists in Python. Please let me know.

Jason Brownlee October 19, 2017 at 3:55 pm #

REPLY ↩

Try it and see.

Erel Segal-Halevi October 19, 2017 at 11:48 pm #

REPLY ↩

Thanks for the post. It may be interesting to check which of these methods can be used for sequence classification.

Method 3 – resampling – cannot be used since, when you undersample or oversample, you harm the sequence structure. I found a paper that relates to this problem:

<http://home.ustc.edu.cn/~zcgong/Paper/Model-Based%20Oversampling%20for%20Imbalanced%20Sequence%20Classification.pdf> but haven't tried it yet.

What about other methods?

Method 5 (different algorithms) – is there a decision-tree variant for sequence classification?

Method 7 (anomaly detection and change detection) – can they be used for sequence classification?

Jason Brownlee October 20, 2017 at 5:36 am #

REPLY ↩

Perhaps you can use data augmentation to create perturbed variations of existing sequences?

Hossein October 24, 2017 at 4:32 pm #

REPLY ↩

Hi Jason! First of all, thank you sincerely for this very useful post that gives a very clear, understandable and informative overview on the imbalanced data.

I developed a fuzzy inference system to detect the errors in the crowdsourced observations (it is very similar to the fraud detection problem). The system predicts the reliability of each observation and assign a value between 0 to 1 to it. So, if the acceptance threshold of reliability is defined as 0.4, all the observations with the reliability value less than 0.4 is considered as error (False).

We know by our experience that almost 95% of observations are correctly classified by the people and just 5% of the observations are misclassified (this is our prior knowledge about the prevalence of error in the dataset).

We defined a 2-class (binary) classification problem (if the observation is correctly detected by the person it belongs to True (T) class, and if it was wrongly classified and it is an error it belongs to False (F) class).

As we developed a fuzzy system, we do not have the training step (as the system makes the decision based on the fuzzy ruleset we defined for it).

However, we need to evaluate the discriminatory performance of our fuzzy system (if it can assign the T and F classes to the observations with the high accuracy or not).

We have only 119 ground truth observations (reference data) for testing the performance of our system. We can feed them to our system and the system is labeling them (T or F), then to assess the performance of our fuzzy system we can cross-check these predicted labels with the real labels of the ground truth observations.

Our groundtruth dataset is imbalanced (114 out of 119 observations are from T class and 5 of them belongs to F class). However, this is very consistent with the prevalence of the error in the dataset (as I mentioned based on our experience we know that usually most of observations are correctly classified and only around 5% of them are errors).

So, I am wondering if we can use this imbalanced (but consistent with the prevalence) groundtruth dataset for evaluation of the predictive performance of my fuzzy system or I HAVE TO resample my 119 groundtruth observations to make a more balance test dataset?

I use the different metrics in my paper to evaluate the performance of my system such as AUC, confusion matrix and Kappa at the different cutoffs (thresholds). I got AUC=98% and the maximum kappa of 0.799.

Thank you so much!

Jason Brownlee October 25, 2017 at 6:38 am #

REPLY ↩

It is hard to say. I would suggest trying some rebalancing techniques on the training set and evaluate the impact on model skill.

Leo Buckley October 27, 2017 at 1:25 am #

REPLY ↩

Hi Jason,

This post is extremely helpful. I have a specific question regarding a dataset that I'm working on. It is patients with heart disease. I am trying to develop an algorithm to predict patients who will have another heart attack. These patients are usually only 5-10% of all patients, but because another event is so devastating, the ability to identify these patients is very important.

I understand that different options are available to improve the predictive capability of the algorithm, but is anything lost when over- or under-sampling the data? My gut says that the distribution of the outcome of

interest (heart attacks) is an important feature inherent to the data that is lost with over- and under-sampling and therefore the predictive model will be very good statistically but not applicable to a broader population of people (biased).

Is this thinking correct or am I missing the point? Thanks!

Jason Brownlee October 27, 2017 at 5:22 am #

REPLY ↩

Yes, the idea is to pervert the distribution with the goal of lifting model skill on the underrepresented case.

Focus on this goal.

Michael November 7, 2017 at 5:24 am #

REPLY ↩

Hi Jason,

1. I understood that AUC (Area Under Curve) is good performance measure for imbalanced data. But, can it be used to measure classification performance for balanced data?
2. I used the AUC with a Majority classifier (ZeroR) and got $AUC < 0.5$. Can I say that the ZeroR classifier is bad?

Thanks,

Jason Brownlee November 7, 2017 at 9:54 am #

REPLY ↩

Yes it can, and ZeroR should be bad, it is the starting point to get better.

Karthi November 17, 2017 at 6:44 am #

REPLY ↩

Useful Post. Thanks...

Jason Brownlee November 17, 2017 at 9:29 am #

REPLY ↩

I'm glad it helped.

Ms S Lalitha November 19, 2017 at 4:03 am #

REPLY ↩

Hi Mr.Jason,

I am working on an imbalanced dataset. Can I use resample technique for cross validation? In case, if this technique is used, will it not create duplicates in the n-fold cross validation? Please reply

Jason Brownlee November 19, 2017 at 11:10 am #

REPLY ↩

Yes, but you must apply any rebalancing on the training set within the cross validation fold/split. It may create duplicate samples, that is the point of some approaches.

Zakia November 21, 2017 at 2:44 am #

REPLY ↩

Hi Jason,

I need a help regarding my experiment in machine learning. I used classbalancer of weka 3.8 to balance my training dataset (100 vulnerable data and 10000 non-vulnerable data). After that I am testing the model on another dataset containing 60 vulnerable data and 2500 non-vulnerable data. The test data is taken from another system (Different from training set). As the test data is not balanced, my precision for test data is very low (less than 1%). But precision of training data is 75%. Even if I consider test data from the same system, it gives low precision. Then I took 100 vulnerable and 100 non-vulnerable data for test which improves the precision. But in real life, test data are generally not balanced. How can I improve my precision on imbalanced test data.

Thanks.

Jason Brownlee November 22, 2017 at 10:43 am #

REPLY ↩

Perhaps there are some ideas here you can try:

<http://machinelearningmastery.com/machine-learning-performance-improvement-cheat-sheet/>

patrick November 25, 2017 at 3:25 pm #

REPLY ↩

Hi Jason,

Thanks a lot for the article. I have a case where the class has three values:

Up

Down

None

But more than 90% of the data has a value of None. I tried under-sampling the data and remove enough of the None values to become balanced. However the issue is that now we are introducing more FP and FN to the results. They appear in the results as soon as I add them back to the data and reuse the trained model.

Is there any way to keep the None values as evidence to prevent false detections but make the model take the Up and Down values as first priorities?

I tried CostSensitiveClassifier in Weka but then it reduces precision or recall.

Really appreciate if you share your thoughts about this.

Thanks!

Shabnam January 4, 2018 at 7:59 am #

REPLY ↩

Thanks Jason for such a great post.

I have two questions:

1. I understood that the accuracy is not trustable when data is not balanced.

How much imbalance is fine? For example, in a binary classification, 1:1 is for balanced data. Should we consider 1:1.5 or 1:2 as imbalanced?

Or it depends on accuracy as well?

For example, if we have 1:2 data and we get 70% accuracy, we cannot trust it, but if it is 90% accuracy, we can.

2. You mentioned about downsampling as one of the methods. Do you have any suggestion for it?

Thanks for your response, time, and help as always.

Jason Brownlee January 4, 2018 at 8:18 am #

REPLY ↩

Accuracy really only makes sense for binary classification with 50/50 split, or very close to it.

Sorry, I don't have any worked examples for downsampling.

Smitha January 14, 2018 at 12:05 am #

REPLY ↩

Hi Jason,

I have a dataset with 2,23,586 samples out of which i used 60% for training and 40% for testing. I used 5 classifiers individually, SVM, LR, decision tree, random forest and boosted decision trees. SVM and LR performed well with close to 0.9 accuracy and recall also 0.9 but tree based classifiers reported an accuracy of 0.6. After a careful observation, I found out that SVM and LR did not predict the labels of 20,357 samples identically. So Can I apply voting and resolve this conflict wrt prediction outcome? Can this conflict be due to an imbalanced dataset? I need your help..

Jason Brownlee January 14, 2018 at 6:37 am #

REPLY ↩

Try it and see.

Smitha January 14, 2018 at 12:06 pm <#>

REPLY 

I Combined SVM and LR to get an accuracy score of 0.99. Recall is 0.98. To resolve conflicts wrt 20,357 samples, I devised voting rule by taking into account all 5 classifiers. In spite of their average performance, tree based classifiers have also contributed towards resolving conflicts being complimentary to SVM and LR. Thank you.

Jason Brownlee January 15, 2018 at 6:55 am <#>

REPLY 

Well done!

Ousama February 2, 2018 at 11:52 pm <#>

REPLY 

Thanks for the great post, your website has always been a great resource for me.. I am currently struggling with a problem where I have around 13 million rows and the targets are binary classes ratio of 6800:1 which is very imbalanced. Moreover, 2000 rows are a class and the rest are another class. Would you think with the factors you specified would I be able to feed it to a classifier or NN?

Jason Brownlee February 3, 2018 at 8:39 am <#>

REPLY 

I would encourage you to explore rebalancing methods regardless of the type of model you choose.

Marlene March 2, 2018 at 9:53 am <#>

REPLY 

Dear Jason, would you have a reference for the 80:20 or 4:1 definition of an imbalanced dataset? I did not see this clearly defined in any references listed. Thank you

Jason Brownlee March 2, 2018 at 3:21 pm <#>

REPLY 

No, sorry.

Amom March 6, 2018 at 10:59 am <#>

REPLY 

Extremely useful! Congrats!

Jason Brownlee March 6, 2018 at 2:56 pm <#>

REPLY 

Thanks, I'm glad to hear that.

Misos March 7, 2018 at 10:05 pm <#>

REPLY 

Hi Jason,

Thanks for the great post.

Could you help listing classifiers which are not affected by Imbalanced classes problem such as KNN please?

Jason Brownlee March 8, 2018 at 6:29 am <#>

REPLY 

Thanks for the suggestion.

hadi sotudeh March 9, 2018 at 9:44 pm <#>

REPLY 

Hi Jason,

Very interesting post.

I have a data set which is very very imbalanced (99.3 percent for the majority class). The minority class has 1 to 2 percent share in all kinds of these data sets I use.

What would be your recommendation for these cases?

Jason Brownlee March 10, 2018 at 6:25 am <#>

REPLY 

Ouch. I would recommend trying everything and see what gets some traction on your specific case.

Some problems are just plain hard and we are happy to get anything better than random chance as a prediction.

Talat CAN March 22, 2018 at 10:14 pm <#>

REPLY 

Thanks a lot!! It helps.

Jason Brownlee March 23, 2018 at 6:07 am <#>

REPLY 


You're welcome.

gini March 30, 2018 at 3:56 am <#>

REPLY 

I have applied the oversampling after the modeling it is possible to correct the probabilities to return to its original distribution

Jason Brownlee March 30, 2018 at 6:45 am <#>

REPLY 


What do you mean exactly?

soeysoo May 13, 2018 at 2:41 am <#>

REPLY 

I had a balance class("YES" and "NO") with 3 attribute (include age, gender, and month). However all the 3 attribute is not balance. For example attribute gender (boy and girl). Boy get 80% "YES" and 20% "NO". Is this also mean i have imbalance dataset although i had a balance class? Should i take it serious in building the classifier?

Jason Brownlee May 13, 2018 at 6:36 am <#>

REPLY 

Perhaps try working with the data as-is, then explore rebalancing methods later to see if you can lift model skill.

Mayank_Satnalika May 21, 2018 at 12:35 am <#>

REPLY 

Hi Jason thank you for your helpful posts.

Say I have a 5:1 unbalanced data, and I'm getting a output probability for the class from my classifier. Would it be a good idea to train 5 different models taking one part of the major class and the complete minor class and finally take the average of them. This would be same as under-sampling but use all available data

because we have 5 models for the 5 different data parts.
Any views on this?

Jason Brownlee May 21, 2018 at 6:31 am <#>

REPLY 

Try it and see how it performs.

Thorbjorn May 24, 2018 at 9:31 pm <#>

REPLY 

Great post – gives a good overview and helps you get startet.

Thanks:-)

Jason Brownlee May 25, 2018 at 9:23 am <#>

REPLY 

Thanks, I'm glad it helped.

Felix June 18, 2018 at 10:11 pm <#>

REPLY 


Thanks a bunch for the great article once again! One question: is it common to resample the whole data set and then make a train-test split, or first split and then resample – testing only on the original data? Does it even make a difference, other than the amount of a given class in the results?

Jason Brownlee June 19, 2018 at 6:33 am <#>

REPLY 

No, resample the training data only.

mmn June 20, 2018 at 2:50 pm <#>

REPLY 

Thanks for a very helpful post! . One question, is the undersampling method useful in highly imbalanced ratio (for example majority : 100 and minority ;5) . ?

Jason Brownlee June 21, 2018 at 6:09 am <#>

REPLY 

It can be. It depends on the problem.

Moyo July 2, 2018 at 6:50 pm #

REPLY ↩

This is best seen when using a simple rule based algorithm. If you print out the rule in the final model you will see that it is very likely predicting one class regardless of the data it is asked to predict.

In the paragraph above you mentioned “printing out the rule of a model”. Please I would like to know how to go about retrieving that information in a model.

Regards

Jason Brownlee July 3, 2018 at 6:24 am #

REPLY ↩

is only makes sense for trees and rule based systems.

HY July 3, 2018 at 4:47 am #

REPLY ↩

Hey Jason, Thanks for sharing the 8 tactics! I was also wondering, will you use the same method when your data is expected to be imbalanced?

Jason Brownlee July 3, 2018 at 6:29 am #

REPLY ↩

These methods are only for when the data is imbalanced.

Brian Tremaine July 14, 2018 at 6:04 am #

REPLY ↩

Hi,

I'm working on a very imbalanced data set (0.3%) and am looking at papers related to credit risk analysis. One such paper that evaluates several sampling schemes is here:

<https://core.ac.uk/download/pdf/61416940.pdf>

Regards,

Brian

Jason Brownlee July 14, 2018 at 6:23 am #

REPLY ↩

Thanks for sharing.

Daniel Cao July 31, 2018 at 12:39 am #

REPLY ↩

Hi Jason,

Thank you so much for your post. At first, I thought balancing the data is a good practice and it helps me with more satisfactory results for many times. However, when I came across the paper King & Zeng paper <http://gking.harvard.edu/files/gking/files/0s.pdf>, it seems that we cannot just solely balance (or bias) the sample. After the training, we still have to “correct” for that bias. In the case of logistic regression, we make the correction using the intercept coefficient.

Have you ever done this in practice, and in case of Neural Network, do we have to do this?

I cannot wrap my head around this issues for weeks.

Thank you.

Sincerely,
Daniel

Jason Brownlee July 31, 2018 at 6:05 am #

REPLY ↩

I'm not familiar with that paper, sorry Daniel.

I often predict probabilities when working with imbalanced datasets, and I've not bias corrected the predicted probabilities other than perhaps a little calibration.

Mira August 17, 2018 at 6:58 pm #

REPLY ↩

Very nice post! Thanks a lot.

Jason Brownlee August 18, 2018 at 5:35 am #

REPLY ↩

Thanks, I'm happy it helped.

Joey Gao August 29, 2018 at 12:11 pm #

REPLY ↩

Hello Jason, In xgboost, 'scale_pos_weight' can deal with imbalanced dataset by giving different weights to different classes, should I do over-sampling or under-sampling before tune 'scale_pos_weight'?

Jason Brownlee August 30, 2018 at 6:21 am #

REPLY ↩

Try them separately would be my advice.

Karanja August 30, 2018 at 8:58 am #

REPLY ↩

hello Jason I wish to undertake low complexity Automated Diagnosis of lung cancer tumor classification for instance which falls in my case in six different tumors classes say (squamous carcinomas,adenocarcinomas etc)

I have a small data set of 200 Images of tumor which can be sub categorized into 6 distinct groups based on Tumor category and 200 (healthy Images) . My aim is

1. To effectively classify the image into its right category say if I have images of tumors from the datasetSuch that provided an image or images I can easily classify within its category.

what is the best classification approach to useIs deep learning ok considering the size of the dataset

Jason Brownlee August 30, 2018 at 4:50 pm #

REPLY ↩

I would encourage you to test a suite of methods to discover what works best for your specific dataset.

I would guess that CNNs would be effective if you are working with images.

Hilda September 27, 2018 at 5:52 am #

REPLY ↩

Hello Jason,

Can we just add noise to the minority class to make synthetic data? what is the benefit of using SMOTE instead of this approach?

Regards,
Hilda

Jason Brownlee September 27, 2018 at 6:06 am #

REPLY ↩

Try it.

Noise is often a good on inputs but not on outputs, mainly to make the mapping function learned by the model artificially smoother and easier to learn.

Anuja Tupe October 17, 2018 at 4:00 pm <#>

REPLY 

Hello Jason!

I am working on some project which is using CNNs. My dataset involves 43 classes and the dataset is highly imbalanced. The highest frequency is 2250 and the lowest frequency is 210.

I read on the web that we should pass class weights to the fit method when you have an imbalanced dataset. By supplying class weights, when the model encounters a training example for a less represented class, it pays more attention and puts greater emphasis when evaluating the loss.

However, I am trying to understand how does the fit method use these class weights. In my model, I have written some custom metric method which calculates fbeta score, precision and recall. I call this custom metric method in callbacks of the fit method. How will the model understand to use these class weights in the custom metric method? Am I supposed to pass class weights to the custom metric method?

I could not find any satisfactory answer for my question. Would really appreciate it if you could help me understand this.

Thank you!

Jason Brownlee October 18, 2018 at 6:24 am <#>

REPLY 

It really depends on the method you're using. In some cases the model can make use of the the weighting, in others only the loss function or evaluation function can make use of the weighting.

Rakesh October 28, 2018 at 1:31 am <#>

REPLY 

Hi Jason,

I always find the solutions to my problems from your post.

Indeed a great article.

Thanks.

Jason Brownlee October 28, 2018 at 6:13 am <#>

REPLY 

Thanks.

Cathy Qian December 6, 2018 at 5:14 am <#>

REPLY 

Hi Jason,

Thanks for the nice post! I wonder what's your criteria for a data set being called imbalanced? For example,

I have a data set with three classes with 10:6:4 ratio and my prediction result gives a prediction accuracy of 80%, 50% and 50% on each class. Do you think my data is imbalanced and that's why I get the highest prediction accuracy on the biggest class? Which of the above methods you mentioned do you suggest I try? Thanks again for a great article.

Jason Brownlee December 6, 2018 at 6:02 am <#>

REPLY 

Any data that is not 50-50 or really close to it is imbalanced.

Try a suite of methods and discover what works best for your specific dataset.

Asis Patra February 13, 2019 at 7:53 pm <#>

REPLY 

This is a great post. very helpful.

Thanks.

Jason Brownlee February 14, 2019 at 8:42 am <#>

REPLY 

Thanks, I'm glad to hear that.

Ashok Narendranath February 25, 2019 at 8:12 pm <#>

REPLY 

Great post!

Jason Brownlee February 26, 2019 at 6:18 am <#>

REPLY 

Thanks.

Damon March 3, 2019 at 2:45 pm <#>

REPLY 

Still a good post after a few years Jason. Very helpful. Thank you.

Jason Brownlee March 4, 2019 at 6:57 am <#>

REPLY 

Thanks.

Sladjana March 5, 2019 at 6:30 pm #

REPLY ↩

Hi Jason,

Thank you for all nice posts. They were really helpful. Do you have any experiences with cost sensitive learning in ANN in Python?

Jason Brownlee March 6, 2019 at 7:45 am #

REPLY ↩

I don't have any posts on the topic, sorry. I hope to cover it in the future.

itisha March 10, 2019 at 5:17 pm #

REPLY ↩

hello sir

i am using semeval-2013 dataset for my project. Its a multiclass classification problem This dataset training and testing data are imbalanced. i am using smote to resample the training data. I want to know whether to resample test data or not?

Jason Brownlee March 11, 2019 at 6:49 am #

REPLY ↩

No, only resample the training dataset.

itisha March 11, 2019 at 5:02 pm #

REPLY ↩

ok, thanku sir.

Also sir, i am using grid search for hyperparameter optimization of SVM classifier. So, i created pipeline for smote and svc and then grid search using 10 fold cross validation. After getting best parameters through grid search, should i retrain the svm with original training set or should i resample the training set again using smote?

Jason Brownlee March 12, 2019 at 6:47 am #

REPLY ↩

I recommend fitting a new final model once you find a top performing set of hyperparameters:

<https://machinelearningmastery.com/train-final-machine-learning-model/>

Sai March 12, 2019 at 1:02 pm #

REPLY ↩

Nice useful post. content is crisp and nice. If anyone interested you go through another blog which I came across recently which has explained various Techniques for dealing this class imbalance problem mostly resampling and penalizing cost function.

<https://knowledgengg.wordpress.com/2019/03/04/this-is-suresh/>

Jason Brownlee March 12, 2019 at 2:32 pm #

REPLY ↩

Thanks for sharing.

itisha March 13, 2019 at 3:21 am #

REPLY ↩

thanku sir

itisha March 13, 2019 at 3:59 am #

REPLY ↩

i have gone through the link you have mentioned for final model.
i have a confusion regarding train of final model on all data.

lets say: i have training (A) nd testing set (B)separately(semantic dataset)

- a) first i do grid search on train data with 10 fold cv to find hyperparameters value.
- b) then after getting tuned parameter.... should i train my final model on only whole training set A or should i train final model on A+B.
- c) then evaluate final model on B.

now doubt is that, if i train final model on A+B,then isn't i am leaking information in in unseen test set B?
My confusion is only training of final model....train on A or A+B....because at the end i have to make predictions on test set B.

Jason Brownlee March 13, 2019 at 8:00 am #

REPLY ↩

Good question.

No need to evaluate the final model. Once you choose an algorithm and parameters, you fit the final model on all available data and start using it.

The idea of choosing an algorithm and set of parameters is to estimate their performance on unseen data. Therefore, you will already have an estimate of the model's performance.

Does that help?

itisha March 14, 2019 at 3:28 am #

REPLY ↩

thanku sir....but still little confusion regarding all available data...does it mean train and test data both?

Also as u said,no need to evaluate final model,,,,but i need to check model performance on unseen test set C. So, how i can do this after fitting final model on all available data.?

I am writing two scenarios from whatever i understood:-

Case 1) let's say A is train dataset, B is dev dataset and c is test dataset independently.

a) gridsearch on train data A for hyperparameter tuning. (10 fold cv)

b) suppose i got C=10 for linearsvc() through gridsearch.

c) now i use C=10 and fit a final model on whole A+B

```
clf=LinearSVC(C=10).fit( )
```

d) lastly i use fitted model clf to predict on unseen test C.

```
clf.predict( C)
```

Case 2) when i have only train set A and test set C. there is no development set separately

a) gridsearch on train data A for hyperparameter tuning. (10 fold cv)

b) suppose i got C=10 for linearsvc() through gridsearch.

c) now i use C=10 and fit a final model on whole A+C

```
clf=LinearSVC(C=10).fit( )
```

d) lastly i use fitted model clf to predict on test C.

```
clf.predict( C)
```

are both cases correct?

Jason Brownlee March 14, 2019 at 9:28 am #

REPLY ↩

There is no one best way, find an approach that makes the most sense for your project.

In general, once a model and config is found, you fit a final model on all available data and use it for making predictions.

Itisha March 14, 2019 at 9:50 am <#>

REPLY 


Ok thanku sir!

Prem Alphonse March 18, 2019 at 4:33 pm <#>

REPLY 

Hi, if the target feature is imbalanced say 2% good to 98% bad, and say 2% is 500 records, what if I use that 500 bad records plus only 500 good records from the 98% and train the model. My Question is will the Model generalize well with that 500 + 500 data as it is 50:50 good vs bad? and I do the selection of that good 500 records based multiple iterations to get the high accuracy as only 1000 records which will run faster in the machine to get the output.

Jason Brownlee March 19, 2019 at 8:52 am <#>

REPLY 

It really depends on the specifics of the data.
Perhaps try it and see?

Prem Alphonse March 19, 2019 at 9:43 am <#>

REPLY 

Thank You Jason

Jason Brownlee March 19, 2019 at 10:46 am <#>

REPLY 

You're welcome.

YUCHUAN CHEN April 12, 2019 at 12:06 am <#>

REPLY 

Thanks for tips, Jason!

As far as I know, these solutions are for classification models, can these tactics be applied to the training of regression models? For example, the trained regression model might perform not very well for the features/data from uncommon scenarios, how can I improve the proportion of the uncommon scenarios during the training stage?

Jason Brownlee April 12, 2019 at 7:48 am <#>

REPLY 

The idea of imbalance does not make sense for regression.

Instead, you have a data distribution that can be transformed, e.g. scaled, power transform, etc.

If you mean rare event forecasting or anomaly detection, e.g. for time series, then this is a whole area of study. I hope to cover it in the future.

YUCHUAN CHEN April 12, 2019 at 8:08 pm #

REPLY ↩

Thank for that Jason.

Mohammed May 4, 2019 at 5:59 pm #

REPLY ↩

Which should be done first? Oversampling the dataset then extract features, or extract features then applying oversampling over these features for every sample in dataset?

Jason Brownlee May 5, 2019 at 6:24 am #

REPLY ↩

Probably the latter, but try both and see what works best for your specific dataset.

Mohammed May 5, 2019 at 8:21 pm #

REPLY ↩

Are there any reference that can cited for this point?

Jason Brownlee May 6, 2019 at 6:47 am #

REPLY ↩

Not off hand, perhaps search papers on the topic?

Andrea Boero May 7, 2019 at 5:16 pm #

REPLY ↩

What about adopting agglomerative clustering hoping to find one or more clusters with an interesting percentage of occurrences of small class?

Thank you

Andrea

Jason Brownlee May 8, 2019 at 6:42 am #

REPLY ↩

Perhaps try it?

Sara May 28, 2019 at 7:45 am #

REPLY ↩

My problem is a binary classification and in my data, each sample data record has 3D Cartesian coordinates (which is not used as attributes for classification) plus attributes used for classification.

In such a spatial data set even if I have equal numbers of two classes, still the classification f1-score gets better or worse by having various 3D spatial distribution of two classes in each train dataset.

Can I call this change of f1-score between different trained models as model variance?

Many Thanks

Jason Brownlee May 28, 2019 at 8:22 am #

REPLY ↩

I'm not sure I understand your question, sorry. Can you elaborate on the question?

Sara May 31, 2019 at 7:38 am #

REPLY ↩

This is a binary classification (0 over 1 classes). Each row of data is a 3D point having three columns of X,Y,Z point coordinates and the rest of columns are attribute values.

Although the classification only uses columns of attribute (features), I keep X,Y,Z at each row(point), because I need them to eventually visualize the results of classification (for example, colorizing points in class 0 as blue while points in class 1 as red).

The points are 3D coordinates of a building, so when I visualize the points I can randomly cut different parts of the building and label them as train/test data. I have randomly cut different parts of building, so I have several train/test data. For each one I do the process (fit model on train data, test on test data and then predict classes of unseen data e.g., another building).

Therefore, I can have different train/test data with different spatial distributions of two classes. (By spatial distributions of two classes, I mean where the two classes are located in the 3D space.)

In train/test data called A, the 3D locations of red and blue classes, are different from those in train/test data called B.

The f1-score of A and B on their test set are different but good (high around 90% for either of classes). However, when I predict unseen data with model fitted to A, the f1-score is awful while when I predict unseen data with model fitted to B, the f1-score is good (and visualizing the building gives meaningful predicted classes).

Can I call this change of f1-score for A and B on unseen data as model variance?

By trial and error, I concluded that when classes 0 and 1 are surrounded by each other (spatial distribution of B) I get good f1-score on unseen data, while when classes 0 and 2 are away from each other I get awful f1-score on unseen data. Is there any scientific reason for this?

For all cases, I have almost equal number of 0 and 1 classes.

Many Thanks

apo May 29, 2019 at 5:50 am #

REPLY ↩

Hello Mr. Brownlee

First of all congrats for this great article. I have read many articles about imbalanced data and I think this is the most completed.

Although, I'm a little confused and I'd appreciate if you could help me. I can't understand the trade off between resampling (which ever technique, oversampling or undersampling) and decrease/increase of threshold.

Let me use one of your above examples, Churn problems. Assuming we have such a classification problem, we know that the class "No churn" or 0 is the majority class and the "Churn" or 1 are the minority. Because of the nature of this problem, we want to have great recall score, the highest the better (correct me if I am wrong).

So, I try two possible solutions:

- 1) I use a classifier (let say LogisticRegression) and I reduce the value of threshold from 0.5 (default) to 0.3. In this way, the recall score is better and precision slightly worse (but still ok) than it was when threshold = 0.5.
- 2) I use oversampling (let say the dataset is small to use undersampling) and specifically SMOTE technique. Again I use LogisticRegression as classifier and I notice that the recall is really good and precision is satisfying(almost the same with precision value when threshold=0.3 in solution 1).

So, which way is more preferable? Is there trade-off between them, if yes which is that?

Thanks in advance

Jason Brownlee May 29, 2019 at 8:58 am #

REPLY ↩

You must choose a method that achieves your project goals. This means thinking hard or talking to stakeholders/domain experts about what is most important, choose a measure that captures that, then pick a method that optimizes your chosen measure/s.

There is no objective best, only subjective best for a given project.

Awal June 5, 2019 at 5:30 pm <#>

REPLY 

Hi Jason, great article.

Do I need to perform oversampling in case of 2:1 data, or it won't make any difference?

Jason Brownlee June 6, 2019 at 6:20 am <#>

REPLY 

Perhaps try it and compare results.

Salomon July 4, 2019 at 3:08 am <#>

REPLY 

Hey Jason, great insight for skewed data sets,
I am working on a Churn model and my data is unbalanced in 16:1 ratio. The data set has only 1300 samples. When I do oversampling with the minority classification I get awesome precision and recall doing cross validation with a random forest model. However, when I try to predict a new data set with the same model both the recall and the precision fall drastically (about 50%). Do you have any idea what might be going on?
Thanks

Jason Brownlee July 4, 2019 at 7:52 am <#>

REPLY 

Perhaps the training dataset or the test dataset are too small or not representative?

khadija July 24, 2019 at 6:45 pm <#>

REPLY 

Hi jason !
i want to learn about the current trends and future plannings in class imbalance deep learning.can you share your opinion ..otherwise i read this article, i learnt alot from it .great effort... Thanks!

Jason Brownlee July 25, 2019 at 7:49 am <#>

REPLY 

Great suggestion, thanks. I hope to cover it in the future.

pnak September 18, 2019 at 2:02 pm <#>

REPLY 

I'm working on image classification (15 classes) and they are imbalanced and i wanted to use smote to balance the data. data which i have is the images not dataset. But i cannot implement smote. can you please me with this problem

Jason Brownlee September 18, 2019 at 2:09 pm #

REPLY ↩

Perhaps try image data augmentation:
<https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>

John September 24, 2019 at 11:28 am #

REPLY ↩

Hello Jason, I was wondering if generating new training data with a DCGAN and then using a classifier algorithm on that training data would be effective for handling an imbalanced image dataset.

Jason Brownlee September 24, 2019 at 1:18 pm #

REPLY ↩

Perhaps try data augmentation methods first.
It could be helpful if your generator can produce very realistic but different images. Try it and see!

Barnett Chiu October 12, 2019 at 6:30 am #

REPLY ↩

Hi Jason,

Is it possible to train a classifier that performs well regardless of the frequency of the minority class? I.e. a classifier relatively robust to the prior.

For example, let's say we've trained a probabilistic classifier that performs well in terms of AUC, for which the training data comprise 40% cases and 60% controls. However the same classifier may not generalize well to a new dataset with, say, 5% cases and 95% controls because, at the least, its probability threshold would not be tuned for such skewed class distribution.

And there are scenarios where it's hard to anticipate the frequency of the minority class beforehand. Let's say we will have to predict new samples in a streaming or incremental fashion, where the minority class frequency remains nebulous, or varies over time (imagine that the data are collected from different geographical regions where prevalence of a target disease changes from place to place).

Well.... I guess the simplest solution would be to train a separate classifier for each geographical region. But it would nice to have a classifier that "on average" performs reasonably well regardless of the percentage of the minority class.

Another solution is to tune the hyperparameters of the classifier wrt to the varying frequencies of the minority class and measure the average performance across different frequencies — and pick the parameters that lead to the best performance in an average sense.

Any other possibilities?

Jason Brownlee October 12, 2019 at 7:12 am #

REPLY ↩

Some models can be insensitive to the class imbalance, and some can be made so (e.g. logistic regression, SVM, decision trees). E.g. setting `class_weight` when fitting some vars to the expected weighting in the train set.

I don't like AUC for imbalanced data, it's misleading:

<https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/>

Streaming is hard. There are good papers and book chapters on this. Without reading them, off the cuff, I'd go for separate models – it's simpler and understandable.

kaka November 29, 2019 at 1:53 pm #

REPLY ↩

It is very good blog. How about time series unbalance data. For example in the e-comercial data, we want to predict the user buy or not a product in next month. we may use the past one year data to predict. But most of time, buy users is just 15% of all users. How to deal with this issue? the method like oversampling or down sampling you mentioned in the blog still work for this. Thanks!

Jason Brownlee November 29, 2019 at 6:18 pm #

REPLY ↩

Thanks!

Great question.

That is not a topic I know a lot about – imbalanced time series classification, sorry. I hope I can cover it in the future.

kaka November 29, 2019 at 7:49 pm #

REPLY ↩

thanks for your response! I hope you can write a blog to solve this issue. This is very helpful to us

chaitanya madaka December 8, 2019 at 1:37 pm #

REPLY ↩

Awesome article. But I wonder if under-sampling will change the nature of the data completely.

chaitanya madaka December 8, 2019 at 1:38 pm #

REPLY ↩

I mean over sampling. ehheh!!!!!!!!!!!!!!

Jason Brownlee December 9, 2019 at 6:44 am #

REPLY ↩

That is the idea, but also the risk.

Sarang December 11, 2019 at 7:35 am #

REPLY ↩

Ohh Dear, you are just gr8.

Every way I read, I'm literally astounded how real are the examples and analogies u have given !!

Jason Brownlee December 11, 2019 at 1:37 pm #

REPLY ↩

Thanks!

AJ Wildridge February 11, 2020 at 3:48 am #

REPLY ↩

Hi Jason,

Looks like you have someone plagiarizing a lot of your work written here...

<https://www.datacamp.com/community/tutorials/diving-deep-imbalanced-data>

Might want to send datacamp an e-mail.

Best,

AJ

Jason Brownlee February 11, 2020 at 5:16 am #

REPLY ↩

Very disappointing, thanks.

This happens all the time now.

Glauber Brito March 7, 2020 at 11:26 pm #

REPLY ↩

Great post. Congrats.

Jason Brownlee March 8, 2020 at 6:11 am #

REPLY ↩

Thanks.

Abdulkarim March 14, 2020 at 6:35 pm #

REPLY ↩

magacayga oo ka bilaabama xarafka A

kana soo jeeda wadanaka Ethiopia

in badan waxan doonayaa in aan bogaaga JASON BROWNLEE wax kabarto
oo aad si gaar ah bogaagan iiga barto balan ma ii qadi kartaa ana waxan ahay qof rabitaankisu yahay sida
adigoo kale in aan dadka u caawiyo xaga waxbarashada

@karim

translate by English

Jason Brownlee March 15, 2020 at 6:12 am #

REPLY ↩

I cannot offer one on one coaching sorry.

You can get started with self-study here:

<https://machinelearningmastery.com/start-here/#getstarted>

Zaily AYUB March 22, 2020 at 12:41 am #

REPLY ↩

Hi Jason

Co incidence, Doc !

i am looking for the information on a treating a imbalance classification especially on the Decision Tree
Techniques.

I will buy your tutorial book as I always supporting your effort.

Thanks.

Jason Brownlee March 22, 2020 at 6:55 am #

REPLY ↩

Thanks!

Volkan Yurtseven May 3, 2020 at 4:08 am #

REPLY ↩

great post, though i have a question. how right to copy the same data in case of imbalance. Isn't this cheating the model. should we trust the results?

Jason Brownlee May 3, 2020 at 6:16 am #

REPLY ↩

As long as the sampling is only applied to the training dataset. It can be very effective.

Perhaps start here:

<https://machinelearningmastery.com/data-sampling-methods-for-imbalanced-classification/>

ola June 11, 2020 at 7:46 am #

REPLY ↩

Great post, though i have a question. About the Near Miss algorithm(under-sampling Technique)!

The basics of the Near Miss algorithm are performed as the following steps:

1. The method begins by calculating the distances between all instances of the majority class and the instances of the minority class.
2. Then, n instances of the majority class that have the smallest distances to those in the minority class are selected.
3. If there are k instances in the minority class, NearMiss will result in $k \times n$ instances of the majority class.

Please, Can you clarify the last point (#3)?

Jason Brownlee June 11, 2020 at 1:28 pm #

REPLY ↩

Thanks.

Good question, the references at the end of this tutorial will help:

<https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/>

Urs July 10, 2020 at 3:06 am #

REPLY ↩

How to handle imbalanced time series classification problems?

Say you're trying to predict stock prices and have a time series of recorded features.

The model should simply classify "buy", "hold", or "sell". Hold decisions largely outnumber buys and sells. So the unbalanced model will predict "hold" for all samples and reach 90+ accuracy.

Samples consist of many timesteps into past and the training classes are determined by looking several timesteps into the future.

The model uses a stateful bidirectional LSTM layer, stateful to benefit from the learning effect of understanding the time sequence.

How can I in this environment over- or undersample? I can't simply remove samples from the batch if I don't want to interrupt the sequence for LSTM.

Any ideas?

Jason Brownlee July 10, 2020 at 6:05 am #

REPLY ↩

Perhaps start with a class weighting.

Urs July 10, 2020 at 7:14 am #

REPLY ↩

tk. yes, did that. used entire data range to come up with the class_weight. Then I select a range of 65000 consecutive samples, build the time sequences (shape 65000,180,15) as one epoch and then train in batches of 256 samples). After each epoch I reset state. Train each epoch 3 times and then select a new (random) consecutive range of 65000 samples for the next epoch.

The result is that I get very diverse training results while going through the different epochs (holding different training data).

But common to all training results is the fact that predictions are always tending in one direction (sometimes towards buy, sometimes towards sell).

Confusion matrix example, could be the other way around with all 0s in first column):

```
[[ 3 17381 0]
 [ 1 96395 0]
 [ 4 17288 0]]
```

classification_report:

precision recall f1-score support

class 0 0.38 0.00 0.00 17384

class 1 0.74 1.00 0.85 96396

class 2 0.00 0.00 0.00 17292

accuracy 0.74 131072

macro avg 0.37 0.33 0.28 131072

weighted avg 0.59 0.74 0.62 131072

The model is not learning and the class_weight does not seem to help.

Any other ideas?

Jason Brownlee July 10, 2020 at 1:41 pm #

REPLY ↩

Nice work.

Try changing the model configuration.

Try changing the model type.

Try alternate framings of the problem.

Try new/different training data.

Try alternate data preparations.

Try diagnosing the cause for poor performance.

More ideas here:

<https://machinelearningmastery.com/start-here/#better>

Urs July 11, 2020 at 1:39 am #

REPLY ↩

investigating undersampling (from imblearn.under_sampling import RandomUnderSampler)

This seems to work for 2D data only. Are there any undersampling algos that allow keeping multi-dim x data?

Jason Brownlee July 11, 2020 at 6:19 am #

REPLY ↩

There may be, I'm not across them sorry. You might have to write custom code.

sid July 24, 2020 at 2:48 pm #

REPLY ↩

Jason,

Big admirer of ur work. But I believe this article is not correct. In fact what RCB (one of the commentor above) was mentioning was right.

But problem with that is, what if my training data which is imbalanced is actually how the real world data is. Then by making it balanced, I am biasing the dataset then.

If your real world data is not what your imbalance dataset depicts (real world is more balanced if you feel), then balancing training data via above methods is useful.

But if your real world data IS imbalanced and main class is rare (like cancer, fraud etc). Then if you use above resampling methods it would give bad results as it biases the training data.

In such cases, either change your cost function to include a measure of prediction cost (multiply cost of wrong prediction for each class P0/P1). Or use class weights directly while training the algorithm (use `class_weight` feature in sklearn etc). As that wud implicitly take care of such class prediction cost.

Have you written in actual code implementation of this in any of your posts? Thanks.

Jason Brownlee July 25, 2020 at 6:09 am #

REPLY ↩

Yes, making the training dataset balanced – biased – can be very effective. It is a field called oversampling:

<https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>

It does not work in all cases, especially if the data has a severe imbalance to the point of outliers vs inliers.

Yes, you can bias the cost function, more here:

<https://machinelearningmastery.com/cost-sensitive-learning-for-imbalanced-classification/>

Yes, I have about 50 tutorials on the topic, start here:

<https://machinelearningmastery.com/start-here/#imbalanced>

And a book on the topic:

<https://machinelearningmastery.com/imbalanced-classification-with-python/>

sid July 24, 2020 at 2:50 pm #

REPLY ↩

See the below article to understand better. And let me know if you have any such posts outlining the implementation of below article. Thanks

<https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28>

Jason Brownlee July 25, 2020 at 6:09 am #

REPLY ↩

Thanks for sharing.

Sid July 28, 2020 at 5:29 am #

REPLY ↩

my point above was that we should not balance the data if reality is imbalanced. While you are saying we should balance it even if it becomes biased.

I am not sure I understand that part.

As per the article I shared above, it says do not balance the data if reality is imbalanced. Use Cost Sensitive Learning methods. Can u comment on that? SMOTE is good, if reality is balanced too but training data got

imbalanced.

If reality is imbalanced, then SMOTE and other sampling techniques should not be used as they would bias the data

Jason Brownlee July 28, 2020 at 6:45 am #

REPLY ↩

Balancing an imbalanced dataset is a known method to improve model skill in some cases.

You can learn more about it here:

<https://machinelearningmastery.com/start-here/#imbalanced>

Use whatever works best for a specific dataset, sometimes it is smote, sometimes it is cost sensitive learning.

sushil August 19, 2020 at 8:17 pm #

REPLY ↩

Hi Jason,

Again an excellent article. I suppose this might be a game saver in my previous mailed post regarding my project with Imbalance dataset. Will work out this approach and try to get the desired results. Many thanks for presenting the concepts and approach in neat and clear manner.

Jason Brownlee August 20, 2020 at 6:41 am #

REPLY ↩

Thank you!

Also, these tutorials may help:

<https://machinelearningmastery.com/start-here/#imbalanced>

Silvia October 29, 2020 at 9:33 pm #

REPLY ↩

Thanks for your post, wonderfull as all of your posts! =)

I've been doing some tests on a NN with an unbalanced dataset, and I decided to mix two approaches: oversampling (SMOTE) & undersampling and weighting the classes (Keras' class_weight).

I find that now my model works far better (using the AUC as metric) for the validation set (which has the original distribution) than for the training set. I don't know if this makes sense or I'm doing something wrong.

Is this a normal behaviour?

Thanks for your help!

Jason Brownlee October 30, 2020 at 6:50 am #

REPLY ↩

Thanks!

Nice.

I would not have expected that, I would have expected worse results.

Luigi January 15, 2021 at 8:28 pm #

REPLY ↩

Hi Jason,

a few model like Logistic regression have an option in their argument called `class_weight=None` by default. In case you set it to 'balanced' that should correct a little bit the decision boundary.

I am not sure whether this option is more effective than Under or Over sampling, what's your opinion about it? I have always known that resampling should be really the last choice..thanks

Jason Brownlee January 16, 2021 at 6:55 am #

REPLY ↩

Correct.

It can help in some cases, in others, resampling the training set will help.

You must discover what works best for your dataset.

m.cihat March 27, 2021 at 12:31 am #

REPLY ↩

Hello

I tried many things but I get very poor results. My dataset contains 450.000 datas with 12 features and a label (0 or 1). My dataset is also imbalanced (1:50). I am using EasyEnsembleClassifier from imblearn library of python. In theory it should work wonders because it creates a subset for each estimator and trains a model for each estimator. It should work right? But what I get is 8% precision, 90% recall. The interesting thing is when I only use one subset(RandomUnderSampler) which contains total of 18.000 datas (9000 class 0, 9000 class 1) it produces exactly same result. What could be the reason of this weird result? Any help is appreciated.

Jason Brownlee March 29, 2021 at 5:53 am #

REPLY ↩

Yes, perhaps you can try some of these methods:

<https://machinelearningmastery.com/framework-for-imbalanced-classification-projects/>

Eddy De Waegeneer April 7, 2021 at 3:56 am #

REPLY ↩

Hi Jason

How to handle a dataset with 128 classes in which a few classes occur either 0 times or only 1 time?

Jason Brownlee April 7, 2021 at 5:13 am #

REPLY ↩

Perhaps you can remove classes with zero or one examples.

A_french_croissant April 12, 2021 at 1:24 pm #

REPLY ↩

In France someone copy paste and translate most of your content

<https://datascientest.com/comment-gerer-les-problemes-de-classification-desequilibree-partie-ii>

Jason Brownlee April 13, 2021 at 6:03 am #

REPLY ↩

That is a shame, thanks for letting me know.

Sourav July 16, 2021 at 11:18 pm #

REPLY ↩

In my case, I am getting good results on Test dataset but very poor on OOT. I am using SMOT to balance the dataset and RF to predict..to give actual numbers its 0.81 AUC for test and .5 fro OOT

Jason Brownlee July 17, 2021 at 5:24 am #

REPLY ↩

Perhaps try some of these ideas:

<https://machinelearningmastery.com/framework-for-imbalanced-classification-projects/>

Gloria July 19, 2021 at 3:40 pm #

REPLY ↩

Hi Jason, very insightful article. Thank you so much.

Jason Brownlee July 20, 2021 at 5:33 am #

REPLY ↩

Thanks!

Mateo Acosta Rojas October 5, 2021 at 11:40 am <#>

REPLY 

You are an incredible educator! You should have one post talking about yourself, about the things you think led you to be this prolific and excellent instructor. 😊

shervin October 7, 2021 at 12:36 am <#>

REPLY 

hi jason

resample is one step in pre-processing or i can do it after feature extraction step?

is there any refrence that i can use it in my thesis for this question?

best regards

Adrian Tam October 7, 2021 at 2:56 am <#>

REPLY 

Either should be fine. Imagine you have a table of features as columns and samples as row. Feature extraction is to manipulate columns, e.g., select some columns, or add column A to column B. Resampling is on rows. Hence these two operations can be done in either order.

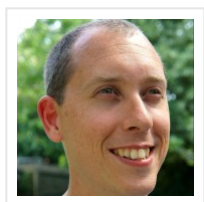
Leave a Reply

 Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT



Welcome!

I'm *Jason Brownlee* PhD

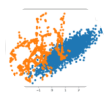
and I **help developers** get results with **machine learning**.

[Read more](#)

Never miss a tutorial:



Picked for you:



[SMOTE for Imbalanced Classification with Python](#)



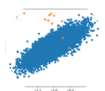
[A Gentle Introduction to Threshold-Moving for Imbalanced Classification](#)



[Imbalanced Classification With Python \(7-Day Mini-Course\)](#)



[Tour of Evaluation Metrics for Imbalanced Classification](#)



[One-Class Classification Algorithms for Imbalanced Datasets](#)

Loving the Tutorials?

The [Imbalanced Classification](#) EBook is where you'll find the ***Really Good*** stuff.

>> SEE WHAT'S INSIDE

© 2021 Machine Learning Mastery. All Rights Reserved.

[LinkedIn](#) | [Twitter](#) | [Facebook](#) | [Newsletter](#) | [RSS](#)

[Privacy](#) | [Disclaimer](#) | [Terms](#) | [Contact](#) | [Sitemap](#) | [Search](#)