Model Building: Best Practices for Imbalanced Data and Partitioning

community.datarobot.com
# Today’s presenters

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matt Marzillo</td>
<td>Customer Facing Data Scientist</td>
<td>Provide guidance and support on framing machine learning problems, data preparation, and modeling activities</td>
</tr>
<tr>
<td>Mitch Carmen</td>
<td>Customer Facing Data Scientist</td>
<td>Provide guidance and support on framing machine learning problems, data preparation, and modeling activities</td>
</tr>
<tr>
<td>Jack Jablonski</td>
<td>AI Success Manager</td>
<td>Create and manage success plan, engagement activities, and resources, serve as your advocate within DataRobot</td>
</tr>
</tbody>
</table>
What is more important?

- Performance at train time
- Performance after deployment

**Trick Question - BOTH!!**
Imbalanced Datasets
Use Cases with Imbalanced Data

- Fraud Waste and Abuse
- Individual Loss Development
- Cybersecurity
- Retail recall predictions
- High Cost Claimants
- Mechanical failures
- Disease Identification
- Customer Churn
Lots of datasets are imbalanced, where the number of observations belonging to one class is significantly lower than those belonging to the other classes.
Multiclass Imbalance

- Not uniform distribution among class categories
- A few classes (or just one class) dominate the number of observations in the data
Zero Inflated Data

- Many observations with zero as target value
- Few observations with target having a wide range of values
Imbalanced Datasets: Conventional Wisdom

While there are more approaches out there, these are some of the standard approaches in data science for handling imbalanced classes.

Warning: It is easy to have leakage/overfit issues if these strategies are not executed correctly. These mistakes are not rare.
1. Collect more data if possible!
   a. Should be first thing we try to make the balance better!

2. Reframe the problem!
   a. **Binary**: Re-define the target to help the imbalance (Ex. Event occurring in 3 mos instead of 1 month)
   b. **MultiClass**: Group similar classes and predict in 2 stages
   c. **Zero-Inflated**: Instead of standard regression problem, set up for 2-stage modeling (Ex. Frequency - Severity modeling)
We optimize objectives that are robust to imbalanced targets (logloss).

Deviance is a measure of goodness of fit compared to a saturated model. Deviance is some measure of the error difference between the target value and the predicted value, where the predicted value is run through a link function. Tweedie Deviance is a more robust link function attempts to differentiate between a variety of distribution families, including Normal, Poisson, Gamma, and some less familiar distributions.

Often great for zero-inflated targets!
Use the Right **Performance Metric**

- Determine correct decision boundary / threshold
- Evaluating AUC and Accuracy is often misleading with imbalanced class problems
- Class 1 precision is usually evaluated, but more minority classes (larger overall dataset) reduces the likelihood that the model found a spurious pattern.

![Diagram showing classification metrics](image-url)
1) To do upsampling, use **weights** to give more emphasis to the minority class

2) Generating synthetic data, like SMOTE has downsides:

   - Computationally intensive
   - Synthetic examples might be misleading - SMOTE assumes the nearest neighbor of that row belongs to the minority class
   - Very easy for novices to create **target leakage**
Downsampling reduces the majority class. This reduces the data for the models to have to process. DataRobot automatically weights the rows and uses a weighted metric to ensure class balance in the dataset.
True Downsampling
Why Downsampling and Upsampling Might Not Help
Use a **Profit Matrix** to Help with Decisions
Partitioning Considerations with Imbalanced Data
Random sampling is the most popular method for assigning observations/rows to a partition.

It is very important not to perform any EDA or feature selection prior to partitioning the Holdout!!!

DataRobot is very fastidious. We don’t include Holdout data in our importance score.
Cross validation is very useful with smaller datasets. Average of these 5 validation scores is the cross validation score. For very small datasets with many features, there are special feature selection techniques (ask your CFDS).
Stratification is the process of rearranging the data so as to ensure that each fold is a good representative of the whole. For example, in a binary classification problem where one class comprises of 20% of the data, it is best to arrange the data such that in every fold, each class is proportionate to the overall distribution.
Nick Roberts quickly identified that there is a leakage issue here. Members of the same group are spread across the train and validation partition.

Were you concerned that the network could memorize patient anatomy since patients cross train and validation? “ChestX-ray14 dataset contains 112,120 frontal-view X-ray images of 30,805 unique patients. We randomly split the entire dataset into 80% training, and 20% validation.”
This leakage can be addressed by partitioning your data by group.

By grouping you learn only from all the observations in specific groups and predict on other groups. Your group feature, here, is the patient.

This allows you to better assess your performance on new patients you have never seen before, and build models more robust to new groups.

To solve it, we hash our anon patient id to assign a k for k-fold CV, ensuring a single patient’s data doesn’t leak across train/test.
The most common partitioning technique is random partitioning.

Ask your CFDS about best practices for partitioning your data.
When a date/time feature is selected in the dataset, it is possible to partition by date and create backtests.
DataRobot will allow you to manually partition the dataset.

You need to use the **Partition Feature** setting and select a column in your dataset that indicates which partition every row should fall into.
Questions & Answers
Engage, learn, and accelerate your AI/ML journey
Connect with peers to find solutions to AI challenges
Explore helpful content to take your AI to the next level
Build your brand as an AI expert & thought leader
Join your peers today at community.datarobot.com
Questions: aisuccess-webinars@datarobot.com
Appendix
Rules of Thumb?

- How many record and positive classes do I need?
  **IT DEPENDS!**
  **But... the more positive observations, the better**

- Evaluating AUC and Accuracy is often misleading with imbalanced class problems
- Class 1 precision is usually evaluated, but more minority classes (larger overall dataset) reduces the likelihood that the model found a spurious pattern.