MFM Datasets Lessons Learned

This is an informal collection of some of the lessons I learned while working on the MFM Datasets project.

Data is King

Machine and Deep Learning algorithms are only as good as the data they are provided, and poor data will invariably cause improper conclusions. In our case, I feel we did a pretty good job of cleaning our variables and performing Extended Data Analysis on most of the fields. However, when the algorithms predicted that Inoxy_incrdose and Intratocolytix were top predictors, it took a long time for us to realize that these were very Site-dependent. I personally could have spent more time understanding the data fields and how their inter-relationships. Crosstab queries can be very helpful for this.

Similarly, cleaning / organizing the data is very important, but must be done somewhat carefully. Some of the ML algorithms, such as SVM (Hsu, 2003), are sensitive to large differences in magnitude. So scaling of the data makes sense in this regard. We may have gone too far, however, in that we removed most of the continuous variables and recast them as Categorical / Ordinal. There were two examples where in the case of the Gradient Boosting algorithm that Continuous variables appeared to outperform similar Ordinal versions: Age & BMI. In later versions of our data, we ran GB with both Continuous and Ordinal versions of these variables, and the Continuous ones were usually higher in the "top predictor" lists. More controlled experiments would need to be run to determine if there is any validity to this trend. One possible confounding factor could be the way these Ordinal variables were defined – the "high_Age" variable, for example was segregated based on common obstetrical age ranges rather than fixed bins. I'm not sure if this could have an effect on the algorithm performance or not.

Missing/Unknown data is difficult

There was a lot of Missing or Unknown fields in our dataset which we handled by imputing to the Median, which is considered one of the best available methods. However, imputing Estimated Bloodloss (EBL) to the median, produced a value of 350 ml, which is "Normal". So, while this did not affect us directly since we used it as a target, some of these women likely had high Bloodloss that wasn't recorded as such. This may have confused the algorithms since this subset were labeled as Negative when they should have been Positive.

By contrast, the Venkatesh paper simply threw out all the data where EBL was missing. This method poses its own set of issues but doesn't suffer from mislabeling.

Alternatively, we could have tried using Machine and Deep Learning methods to impute Missing data. There are examples of using both Autoencoders (Suvra Jyoti Choudhury, 2019) and simple Multilayer Perceptrons (José M. Jerez, 2010) to impute missing data.

More data isn't always helpful

In general, ML and DL algorithms produce better predictions when they have more data to work with. We found a couple exceptions in our case, however.

- Surprisingly, the Gradient Boosting algorithm usually performed somewhat better when using undersampling (1:1) as compared to class-based, sample weighting, which used all the data but with the minority class weighted higher than the majority class.
- Oversampling, specifically using the SMOTE algorithm, did not produce better predictions than class weighting. We did not try a combination of random under and oversampling, however, which may produce better results.
- When creating the trans_loss target (see below) we combined Transfusion with Estimated Bloodloss (EBL) to increase the number positive cases. However, our prediction results were worse that EBL alone for reasons described below.

Choose target carefully

The choice of the prediction target is as important as data preparation. And the clinical significance of a target variable, while a good starting point, should be revisited often. In our case, the transfus_yes variable, while very important clinically, doesn't appear to work as a target variable in this dataset because of the institutional differences in when transfusion is performed.

Multipart targets are especially tricky

Care must be taken when using multipart, or combined variables as the target. In our case, the "trans_loss" variable was defined as "transfus_yes" OR "Estimated Bloodloss >= 1000ml". In the same manner, "transfus_yes" itself was defined as "Postransfus" OR "Bloodproduct". Combining these three variables in this way, while reasonable, does require a strong understanding of what the actual clinical practice is in defining them and how well they are coded by the institutions. I'm not sure we had enough data to make an informed decision in this case. The intra-site experiments seem to indicate a wide variation in how and when transfusion is performed.

Complicating this target selection was the fact that all three of these variables had large numbers of Missing/Unknown values (see above discussion on Missing/Unknown Data). While this doesn't eliminate these fields from consideration as targets, it probably requires more statistical analysis to make valid conclusions.

Finally, we didn't run any of our models with just the "Estimated Bloodloss" portion of our target until very late in the process. With a multipart target like this, it makes sense to run the algorithms against the individual parts to see how they each perform individually. In our case, we might have seen that EBL was performing very well and that the "transfus_yes" portion of "trans_loss" was dragging it down.

Combining sites can produce sub-group affects

It appears that sub-group affects, namely the Sitenum in our case, caused major problems with our dataset. This seems like a difficult problem to solve in general since we need as much data as possible, but combining multiple sites can introduce these sub-group issues. This possibly could have been handled earlier in the process with some cross-site extended data analysis. We may have seen the large

disparity in the application of transfusion between the different sites and therefore avoided that target or controlled for Site in some manner.

It seems like this could be an issue for any analysis using the CSL dataset. And it could extend to many other clinical studies that make use of data from multiple sites without having consistent standards for when certain procedures are performed.

Receiver Operating Characteristic Area Under the Curve

Use multiple statistics for imbalanced data

Our chosen targets were all highly imbalanced, with a positive to negative ratio of at least 14:1. The traditional use of Accuracy and ROC_AUC (Receiver Operating Characteristic Area Under the Curve) can be very misleading in the case of imbalanced data. While these statistics can still be useful for comparison purposes and as a cross-check, the following statistics that emphasis the predictive performance of the Positive class should be favored.

- Precision Recall Area Under the Curve (PR_AUC) : Similar to ROC_AUC, this statistic is more consistent than ROC_AUC for imbalanced data (Saito & Rehmsmeier, 2015).
- Matthews Correlation Coefficient (MCC): A calculation that used all quadrants of the Confusion Matrix, this statistic is often cited as the most robust for handling imbalanced data (Chicco, 2020).
- Precision/Recall/Specificity: These statistics calculate different portions of the Confusion Matrix and provide insight into the quality of the prediction. Recall/Sensitivity is especially useful for imbalanced dataset since it is sensitive to False-Negatives which can be critical in a clinical setting.
- F1, F2, F-beta: The F1 Score is the harmonic mean of the precision and recall. F-beta is a generalized version that applies a weighing factor (Beta) to skew results closer to recall (e.g., F2) or precision (e.g., F 0.5). In the case of imbalanced data with fewer positives, F2 likely is preferred.

Also, the clinical “cost” of miss-prediction should also be considered when evaluating predictions results for a specific statistic. In our case, the “cost” of a False Negative where a woman is at high risk for an adverse outcome isn’t flagged could be much more detrimental than one who is flagged as a False Positive and given special care that isn’t necessary. Obviously, this depends on the specific intervention used.

Algorithms

Both Gradient Boosting (GB) and Random Forest (RF) perform well against this dataset. With additional hyperparameter tuning, I believe that Support Vector Machines could also be improved, although at a significant running time penalty. The Deep Learning examples also have a great potential to be improved, and may even outclass GB and RF. (See more below).

It may have been worthwhile to try some of the other Gradient Boosting algorithms, namely Extreme Gradient Boosting (xgboost). LightGBM and CatBoost. At least in a partial run, LightGBM appeared to perform on par or slightly better than standard GB.

Hyperparameter / algorithm tuning

When tuning ML hyperparameters, start with defaults

Many of the machine learning algorithms, especially the best-performing ensemble, tree-based algorithms such as Gradient Boosting and Random Forest, produce reasonable results with the default hyperparameters. This is dependent on the quality of the input data, however. Also, the defaults can run much slower due to the algorithm needing to work harder to find a minimum. However, the default are often a better starting point than trying to guess which hyperparameters to start tuning.

Be careful of stochastic variation

Machine and Deep Learning algorithms can be sensitive to random/stochastic variation. A couple techniques to ameliorate this include cross-validation and averaging over many initial random seeds. This is especially useful in identifying top predictor variables given their sensitivity to this issue.

Using Optuna (or other) can save effort if done correctly

Using a third-party algorithm tuning application (e.g. Optuna) (Optuna: A Hyperparameter Tuning Framework, n.d.) can both save time and produce a better result. It can be coded to try multiple algorithms (both machine and deep learning), perform hyperparameter tuning and reduce stochastic variability by running cross-validation or multiple runs with different random seeds. While the initial learning curve can be a bit steep, use of this type of tool can both help optimize the algorithm and increase confidence in the results.

"Top Predictor" variables are difficult to assess

- Predictions are based on specific model and vary per run
- Need to run with multiple random seeds like when determining the Confidence Interval.
- Use of Shapely values using the SHAP software library (SHAP, n.d.) can provide insight into which variables are chiefly responsible for a specific prediction run.

Specific MFM comments

Not clear that feature selection helps in our case

We initially spent a lot of time performing feature selection using Cramer and Theil while also trying to use cohorts. While this was a useful in that it provided insight into the associations among our variables, and was a good cross-check against our results, our top algorithms performed better when using all the data fields. In our case, this was not an especially high number (191 for the Pre-Intra set), and did not require high compute resources. A different dataset may very well need this type of feature selection to keep the compute time reasonable.

Deep Learning needs additional research

The two Deep Learning networks we tried produced middling results: better than Logistic Regression, but worse than Random Forest and Gradient Boosting. However, due to the higher coding requirements and lack of time, we only scratched the surface of what is possible. The first thing to do would be better

tuning of the neural network hyperparameters, such as number of layers, neurons per layer, activation function, loss function, number of epochs and learning rate. Very little was done in this regard and could greatly improve the prediction performance. At the time I did not know that a tool like Optuna can be used to automate the tuning of Deep Learning algorithms and could save quite a bit of time.

Consider reproducing other results as a starting point

In hindsight, I believe it may have been worthwhile to attempt to reproduce the Venkatesh results on our own given the similarity of our goals. The advantages of starting with this approach, if successful, would have provided us with a reasonable first-cut from which to build. It may also have reduced and shortened our learning curve with this data. From there we could have branched off and experimented with everything we did, but it would have provided a point of comparison.

The negatives of this starting approach are that it would have been difficult completely reproduce the results given the data we had, and it may have caused us to "lock in" on their method of analysis. It also may have been more difficult to publish since, to me, there seems to be an over-emphasis to produce "novel" results.

Conclusion

Overall, this was a fascinating and useful project to work on. While we were not able to identify a specific model and feature set that could be used to definitively identify women at risk of severe consequences, I do feel our work can be used to move the ball forward. With a change of target, and a few additional data cleaning steps, I believe that one of the Gradient Boosting algorithms could produce a model that replicates across datasets. Also, with concerted use of cross-validation and multiple, averaged random-seed runs, we could find as set of top predictor variables that could be used to quickly ascertain the risk of hemorrhage in a clinical setting. Furthermore, the use of neural networks (Deep Learning) should be explored more to see if they could produce even better predictions.

Bibliography

Chicco, D. J. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics, 21*(6).

Hsu, C. W. (2003). *A practical guide to support vector classification.* Retrieved from Data Science Association.

José M. Jerez, I. M.-L. (2010). Missing data imputation using statistical and machine learning methods in a real breast cancer problem,. *Artificial Intelligence in Medicine*, 105-115.

Optuna: A Hyperparameter Tuning Framework. (n.d.). Retrieved from https://optuna.org

Saito, T., & Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS One, 10*(3).

SHAP. (n.d.). Retrieved from https://shap.readthedocs.io/en/latest/index.html

Suvra Jyoti Choudhury, N. R. (2019). Imputation of missing data with neural networks for classification. *Knowledge-Based Systems, 182*(104838).