Experiment in Logistic Regression Model Tuning

to Decrease False Negative Rate

In an effort to better understand the implications of precision vs. recall and false positives vs. false negatives in machine learning algorithms we conducted a brief experiment using a logistic regression model. Logistic Regression did not turn out to be the most effective model overall but it allowed us to better conceptualize some of the mechanics of machine learning algorithms.

In this context our model is designed to predict whether a patient may be at risk for diabetes.

We came to the conclusion that the benefit of reducing the false negative rate may well outweigh the cost of increasing false positives.

One could make the argument that telling a patient they may be at risk for diabetes when they really are not (false positive) is **less bad** than telling a patient they are not at risk for diabetes when they actually are (false negative).

To reduce the false negative rate we adjusted a few simple parameters in the logistic regression model.

* **The C Parameter**

The C parameter controls the amount of regularization in the model. This controls the trade-off between fitting the training data accurately and allowing the model to generalize better with new data. It can help to reduce overfitting.

Large ‘C’ value means less regularization and therefore a higher chance of overfitting

Small ‘C’ value means more regularization and therefore better generalization but possible underfitting.

* **Class Weight**

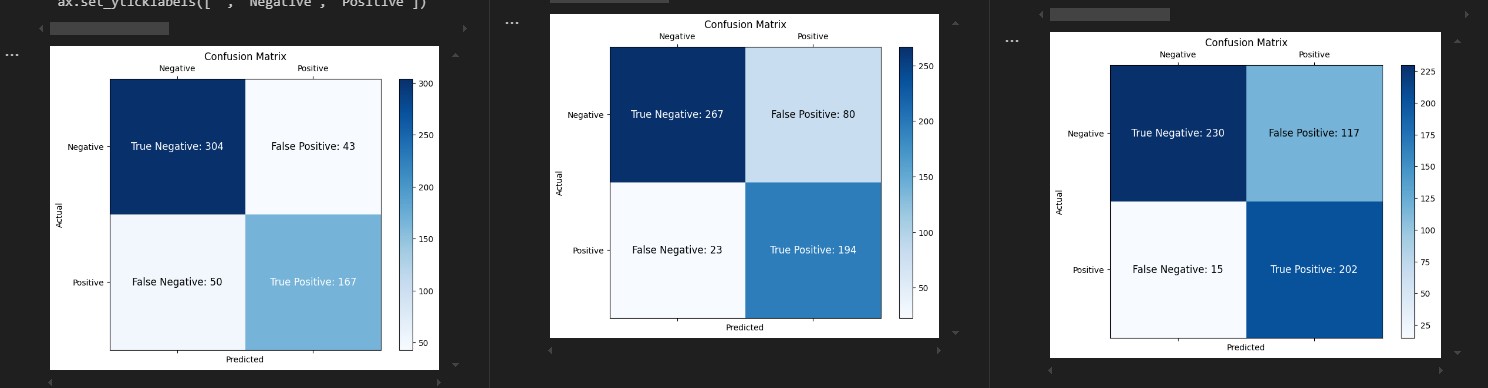
The class weight parameter is used to deal with imbalanced datasets where the classes have unequal representation. This allows the model to assign different importance or weight to each class making the model more sensitive to the minority class in the dataset.

When a model is imbalanced in the target class (for example 90% Class A and 10% Class B) it can become biased toward the majority class and cause poor performance in predicting the minority class when appropriate to do so.

Grid Search

Grid search is a technique used in machine learning that can help to find the optimal and best parameters for a model based on your chosen metric (accuracy, precision, recall, etc.) In this case we used grid search to find the best parameters for recall in order to help reduce the false negative rate.

By doing so we were able to reduce the false negative rate by 70%. However this came at the cost of a significant rise in false positives (174% increase) and a reduction in accuracy score (0.83 to 0.73 respectively).



Precision Vs. Recall

* Precision answers the question, “Of all the positive predictions, how many were actually correct?” High Precision indicates a low false positive rate.
* Recall answers the question, “Of all the actual positives, how many did the model correctly identify?” High recall indicates a low false negative rate.
* A good model should have a Precision-Recall curve that bows upward toward the top-right corner. This shows that the model has both high precision and high recall.

Trade-Off Between Precision and Recall

* **High recall, low precision**: As you increase recall, the model is more sensitive and identifies more true positives, but it might also increase the number of false positives.
* **High Precision, Low Recall**: If you focus on improving precision, the model makes fewer false positive errors, but it might miss some true positives.

The ROC Curve

The ROC (Receiver Operating Characteristic) curve is a graphical way to evaluate the performance of a binary classification model. It plots the **True Positive Rate** (TPR) against the **False Positive Rate** (FPR) at different levels of threshold.

The term ROC (Receiver Operating Characteristic) traces its origins back to signal detection theory developed during World War II. This theory was developed during research in radar and used mathematical concepts borrowed from statistics. It helps to specify the optimal observation and decision processes for detecting electronic signals against a background of random noise. It short it helps to distinguish signal from noise.

Interpretation

* The ROC curve being in the upper left indicates that the model is performing well.
* This means the model has high sensitivity (True Positive Rate) with a low False Positive Rate.
* A perfect classifier would have a curve that goes straight up to TPR = 1
* A model that performs no better than random guessing would produce a diagonal line form the bottom left to the top right corner.

AUC (Area Under Curve)

* AUC is calculated on a scale of 0 to 1.
* AUC = 1: A perfect classifier
* AUC = 0.5: No better than random guessing (the diagonal line in the ROC Plot)
* AUC < 0.5: Worse than random guessing (the model is inverted in its predictions).

The ROC Curve in Practice

* **Threshold Selection:** By analyzing the ROC Curve, you can determine a threshold value that is balanced between True Positives and False Positives based on your specific needs. If you want to reduce false positives you could select a threshold where the FPR is still low and the TPR is relatively high (close to top left corner).
* **Comparing Models:** The ROC Curve is useful for comparing models. A model with a higher AUC curve is generally better at classifying positive and negative classes.