Jason “Scott” Person

Learning Response 4.4: Advanced Churn Modeling

20250501

# Overview

Having geared up your understanding of advanced feature engineering techniques and model evaluation, let’s see how these steps have been put to use by others in modeling two churn data sets.

# Required Learning Resources

1. **Utterback, Predict Customer Churn With Precision – Towards Data Science**  
   <https://towardsdatascience.com/predict-customer-churn-with-precision-56932ae0e5e3>  
   ***Plus* Utterback’s Jupyter notebook:**<https://github.com/cutterback/p03-telco-churn-model/blob/master/Telco-Churn-Classification-Model.ipynb>
2. **Silva, Predicting and Preventing the Churn of High Value Customers Using Machine Learning** – Towards Data Science  
   <https://towardsdatascience.com/predicting-and-preventing-the-churn-of-high-value-customers-using-machine-learning-adbb4a61095d>   
   ***Plus* Silva’s Jupyter notebooks:**
   * Churn Data Wrangling and EDA - Github<https://github.com/jeremysilva1098/predicting_churn/blob/master/Predicting%20Churn-%20Wrangling%20and%20EDA.ipynb>
   * Churn Modeling – Github  
     <https://github.com/jeremysilva1098/predicting_churn/blob/master/Predicting%20Churn-%20Modeling%20Iterations.ipynb>

# Response Questions

### From Utterback, Predict Customer Churn with Precision + Jupyter Notebook

NOTE: Utterback uses the IBM Telco Churn data for his example — so his data should look familiar.

1. Utterback suggests that this churn problem requires that we pay more attention to the recall of our models, rather than simply accuracy. Why? What value would higher recall have for business purposes?

I’m going to go rogue here and say that I disagree with the premise of the question. Utterback states that “we need confidence in our positive class predictions (churn) when taking retention actions.” With respect to statistics, this means that we want high precision – we want high confidence that those who are predicted to be churners are in fact high probability churners.

If we are strictly comparing recall to accuracy then yes, in this case Utterback prefers recall – we want to make sure that we identify all the churners.

From a business perspective higher recall means that the potential churners are more likely to be identified. Those folks will be targeted in our retention efforts.

1. Utterback uses the ROC curve AUC score (Area Under the Curve) to evaluate his models. Given the business goals he describes, what advantage does this offer for building a more valuable model?

ROC AUC allows Utterback to optimize true and false positive rates. A curve which achieves a higher true positive rate with comparatively lower false positive rate will have a greater area underneath it. Further, this curve will facilitate the analysis of decision threshold where we balance precision and recall and can find the point where we get appropriate value from identifying churners without erroneously flagging an excessive number of customers who otherwise would not churn.

1. How does Utterback address the class imbalance of the target variable? (Note: He does not use SMOTE, though he could have.)

Utterback uses the scale\_pos\_weight parameter of the XGBoost binary classification algorithm to put 2.8x greater emphasis on the minority (positive) class. This is based on the positive churners being 2.8x less prevalent in the set than the negative class. He arrives at this 2.8 value with the following equation:

pos\_weight = round((y\_train.shape[0]-np.sum(y\_train, axis = 0)) / np.sum(y\_train, axis = 0), 1)

This takes the total size of the set minus the size of the positives and then divides this by the size of the positives. Effectively this is number of negatives divided by number of positives, which is 2.8. I investigated why he didn’t just use a <some parameter> = “balanced” technique like he did for other algorithms. Apparently XGBoost does not support this so you must calculate the weighting manually. More investigation required here.

1. Which algorithm produced Utterback’s best model? And what performance did the model achieve in AUC, precision, and recall?

Utterback states that his best model uses the XGBoost algorithm. This had AUC of 0.79, precision = 0.54, and recall = 0.83. Notably, the precision can be increased significantly by sacrificing some recall.

### From Silva, Predicting and Preventing the Churn of High Value Customers + Jupyter Notebook

NOTE: Silva uses a different churn data set, but many of his machine learning steps are directly relevant to a data set such as the IBM Telco data set as well.

1. Summarize Silva’s perspective on the comparative importance of accuracy, precision, and recall in relation to this data set. What algorithm produced Silva’s best model, and what performance was it able to achieve with these measures (accuracy, precision, and recall)?

Silva wants to miss as few actual churn cases as possible. He states, “there are no significant consequences of identifying a customer as a Churn risk when she isn’t.” This indicates that Silva prefers recall over precision. As with most churn discussion, in this case accuracy takes a back seat due to the minority class being proportionally small relative to the set size.

Silva settled on a Random Forest Classifier backed by SMOTE. With this algorithm, Silva’s model was 91% accurate, had a precision of 0.84 and a recall of 0.91.

1. Why does Silva like the F1 Score? What are its advantages? And what score was he able to achieve with this model?

Silva states that “The F1 Score helps keep us honest.” By this he means that it forces a tradeoff between precision and recall. Because the F1 Score is the harmonic mean of recall and precision, if one of these is quite low, the value of F1 would be significantly penalized. Disproportionally emphasizing one will result in the other going down. The final model had an F1 score of 0.87.

1. Summarize Silva’s use of SMOTE, including the proportion of Churn to Non-Churn cases he winds up using, and his statement about the contribution of this step to the performance of his models.

Silva uses SMOTE to generate additional records that have a higher prevalence of churn. The training set contains 1 churn record for every 5.5 non-churn records. Note that there is an apparent typo in the article so I’m assuming that this is 1:5.5 and not 1 out of 5.5, which would be 1:4.5. Regardless there are relatively few churn records compared to the training set as a whole. SMOTE added synthetic records until the churn to non-churn ratio was 1:2.

Silva states “This helped improve performance greatly.” Examination of the model comparison grid indicates that SMOTE added approximately four percentage points to the F1 Score.