Week 6 - Build Candidate Models

Over the previous two weeks, you built a good baseline model for prediction. Now you have a point of comparison for more sophisticated approaches which we expect to provide better performance, though often at the cost of higher computational complexity. This week, each member of the capstone team is going to build their own classifier with a different algorithm. Use the previous weeks' instruction to guide you in the process.

Here are some options to get you started.

- Naive Bayes
- K-Nearest Neighbor
- Decision Tree
- Random Forest
- Gradient Boosting
- XGBoost

At this point you have a candidate model for predicting whether someone will subscribe to the term deposit. Now you need to use the data to tune hyperparameters with that model. Follow the same process that you used to tune the L2 Logistic Regression with grid search and k-fold cross-validation; however, your models might have multiple parameters that you should consider tuning. Be careful with adding too many parameters to the grid search because it will search every combination of parameter value that you provide. If you decide to search over multiple parameters and many values for each parameter, consider using a random search.

Note: Hyperparameters are not constrained to arguments that you explicitly pass to your model. They also include data pre-processing, resampling (e.g. SMOTE), and any other transformation you make on the data. We recommend that you treate resampling as a hyperparameter. To simplify this process, produce a tuned candidate model with and without resampling to be evaluated in model selection next week.