BNP Paribas Cardif Claims Management

A Data Story *for* Data Science Intensive

Introduction

For my Springboard Data Science Intensive Capstone project, I selected a recently concluded Kaggle competition, sponsored by BNP Paribas Cardif, which is the insurance arm of BNP Paribas. This competition, titled BNP Paribas Cardif Claims Management, challenged Kaggle members to accelerate their claims management process by predicting if a claim:

a. Could be accelerated for approval leading to faster payments, or
b. Requires additional information before approval is made

In their own words, this is an important problem to solve because:

"In a world shaped by the emergence of new uses and lifestyles, everything is going faster and faster. When facing unexpected events, customers expect their insurer to support them as soon as possible."

Provided Data

The organizers provided two datasets:

1. Training data with a binary target (0/1 where 1 indicates that the claim is suitable for accelerated approval)
2. Test data of the same form, obviously without target values

Both datasets have ~114,000 rows, where each row corresponds to a claim, and both have 131 features/predictors. The features are both numerical and categorical, **and they contain the information available to BNP Paribas Cardif when the claims were received**. Also, there are no ordinal variables, meaning that none of the categorical variables have an order embedded in the values taken by that variable.

Finally, a unique feature of the datasets is that all of the 131 features are anonymized, i.e. we have no information about what they represent. The features are numbered v1 through v131.

The Challenge

Submissions to the competition were judged based on the Log Loss of the model on the test dataset:

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i))$$

where N is the number of observations, log is the natural logarithm, y_i is the binary target and p_i is the predicted probability that y_i is equal to 1.

Exploratory Data Analysis

I used IPython notebooks for my analysis. The notebook with the code and comments is available in the Github folder where all the other DSI documents are located (https://github.com/rr2/DSI).