Bosch Production Line Performance

Oliver-Matis Lill

Introduction

- We have measurements from the production line and the goal is to predict failures
- 6GB training and test sets
- About 4000 columns and 1,200,000 lines
- The columns are either categorical or numeric
- The columns names are anonymized, human-level analysis is difficult
- The data is very unbalanced, training set has 1,176,868 sucesses 6879 failures
- We want to maximize Matthews Correlation Coefficient:

$$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

- The format is pretty simple:

```
Id,L0_S0_F0,L0_S0_F2,L0_S0_F4,L0_S0_F6,Response
122,0.049,0.041,0.385,0.384,0
694, , , , ,0
5340,0.062, ,-0.161, ,1
```

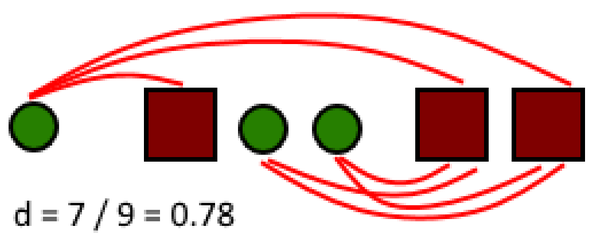
Numeric Relevance Measure

- Suppose that in successes the values are $u_1, u_2, \ldots, u_m$ and in failures the values are $v_1, v_2, \ldots, v_n$
- The failure displacement d is the number of (u_i, v_j) pairs where $u_i < v_j$, divided by all possible (u_i, v_j) pairs:

$$d = \frac{|\{(u_i, v_j) | u_i < v_j\}|}{nm}$$

- The relevance measure is:

$$R = |d \cdot 2 - 1|$$



Ideas

- We could pick a smaller subset of more relevant columns
- We need to find a way to estimate a relevance of columns
- We also need the relevance measures of categorical and numeric columns to be comparable
- Maximizing MCC roughly corresponds to maximizing accuracy in a balanced dataset
- For performance reasons it's a good idea to use undersampling, the patterns in overrepresented outcome will likely remain detectable

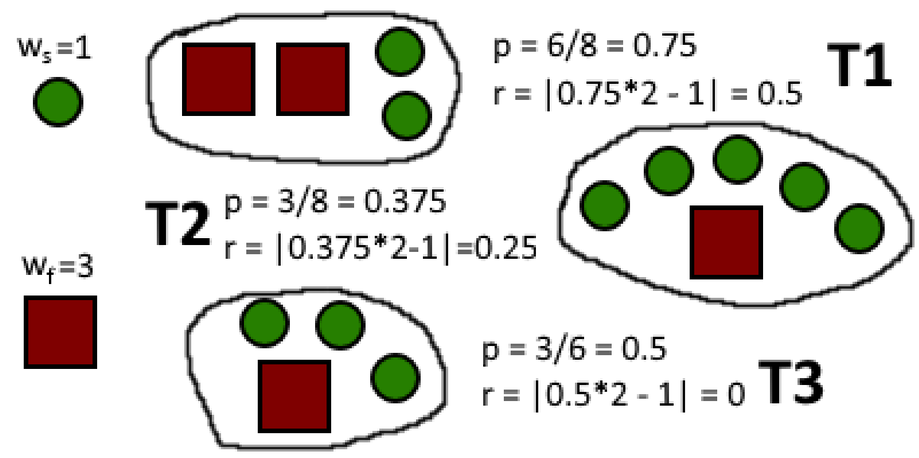
Categorical Relevance Measure

- Assign success with weight w_s and failure with weight w_f
- If a category has a successful products and b failures, its proportional failure is:

$$p = b \cdot w_f / (a \cdot w_s + b \cdot w_f)$$

- Proportional failure can be used as a numeric value of category. The relevance measure is:

$$r = |p \cdot 2 - 1|$$



Modeling

- Even with subsampling, the dataset sizes turned out too much for R to handle, so I had to turn to python
- I some tried simple neural network architectures in Python using Keras
- In the end the highest validation accuracy I managed was around 64%, which is not high considering there were only 2 classes
- In the end I suspect there is not much much more I could have done with models. Probably even more effort is needed in preprocessing and feature extraction