Email sorting with Natural Language Processing and Conformal Prediction

Problem
- Automatically forward customer email to the right agents
- Based on message content
- Control the number of errors

Data

Sparse Text Vectors
\[ d_1 = \text{“my bill is too high”} \]
\[ d_2 = \text{“last bill was a high bill”} \]

Feature scaling
- Term Frequency \( t f(t, d_i) = \# t \in d_i \)
- Document Frequency \( df(t) = \# d : t \in d \)
- \( \text{TF-IDF}(t, d_i) = t f(t, d_i) \cdot \log \left( \frac{n}{df(t)} \right) \)
- TF-IDF score penalizes terms that appear in too many documents

Final dataset: matrix \( M \in \mathbb{R}^{40,000 \times 7,500} \)

Text pre-processing
I've already paid £150 on 10/03/2019. Why the new bill?

We undersampled the largest 7 classes to have 5,000 examples per class

Performance
- Random Forest’s prediction region is the most efficient
- Apart from KNN, models produce correct single predictions 70% of times
- Slowest: SVM (6+ hours) and KNN
- Recommended: Random Forest (1000 trees)

Conclusion & Future work
- Good result given the limitations:
  - Undersampled dataset
  - Several wrongly labelled examples
  - We can decide if a human intervention is needed in each case
  - Will use Mondrian predictors for imbalanced classes
  - Will use dense embeddings and deep neural network as underlying algorithms

References
- Linusson, H. nonconformist Python module