Applied Machine Learning

Feature Selection: Filter Methods

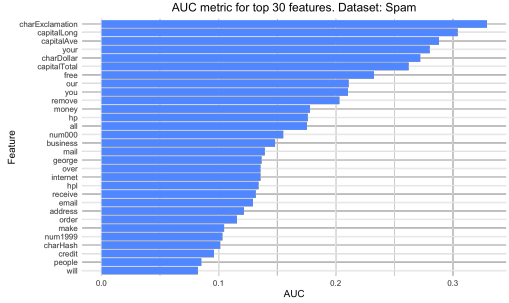
Learning goals

- Correlation-based Filtering
- AUC/ROC-based Filtering

Filtering

INTRODUCTION

- **Filter methods** construct a measure that quantifies the dependency between features and the target variable
- They yield a numerical score for each feature x_j, according to which we rank the features
- They are model-agnostic and can be applied generically



Exemplary filter score ranking for Spam data

PEARSON & SPEARMAN CORRELATION

Pearson correlation $r(x_j, y)$:

- For numeric features and targets only
- Measures linear dependency
- $r(x_j, y) = \frac{\sum_{i=1}^{n}(x_j^{(i)} - \bar{x}_j)(y^{(i)} - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_j^{(i)} - \bar{x}_j)}\sqrt{(\sum_{i=1}^{n} y^{(i)} - \bar{y})}}, \qquad -1 \leq r \leq 1$

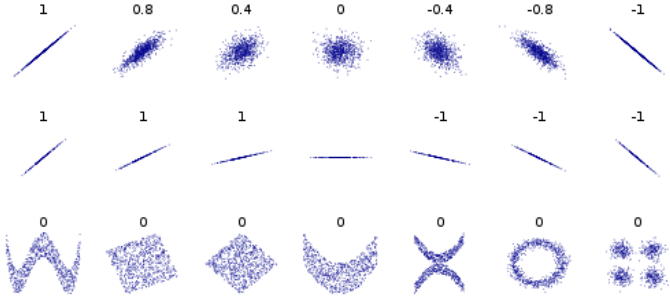
Spearman correlation $r_{SP}(x_j, y)$:

- For features and targets at least on ordinal scale
- Equivalent to Pearson correlation computed on ranks
- Assesses monotonicity of relationship

Use absolute values $|r(x_j, y)|$ for feature ranking:
higher score indicates a higher relevance

PEARSON & SPEARMAN CORRELATION

Only **linear** dependency structure, non-linear (non-monotonic) aspects are not captured:

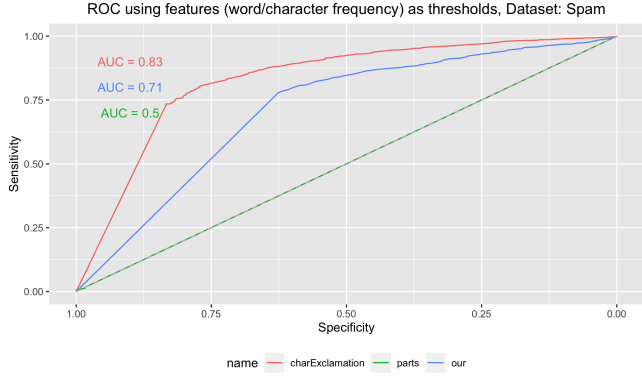


Comparison of Pearson correlation for different dependency structures.

To assess strength of non-linear/non-monotonic dependencies, generalizations such as **distance correlation** can be used.

AUC/ROC

- For binary classification with $\mathcal{Y} = \{0, 1\}$ and numeric features
- Classify samples using single feature (with thresholds), compute AUC per feature as proxy for its ability to separate classes
- Features are then ranked; higher AUC scores $\rightarrow$ higher relevance.



FURTHER CRITERIA/METHODS

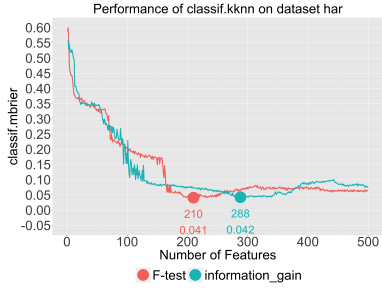
- Classification
  - χ^2-statistic
  - Welch's t-Test
  - F-Test
- Regression & Classification
  - Mutual Information
  - Relief
- Model-based
  - Tree-based feature importance (decision tree, extra trees, random forest)
  - Coefficients of linear model

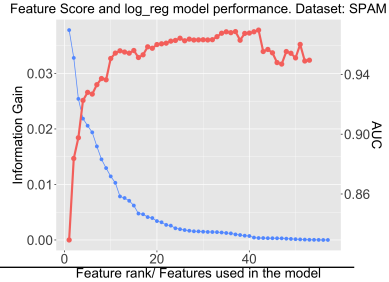
USING FILTER METHODS

1. Calculate filter score for each feature x_j
2. Rank features according to score values
3. Choose $\tilde{p}$ best features
4. Train model on $\tilde{p}$ best features

How to choose $\tilde{p}$?

- Could be prescribed by application
- Eyeball estimation: read from filter plots
- Treat as hyperparameter and tune in a pipeline, based on resampling





ADVANTAGES AND DISADVANTAGES

Advantages

- Fast
- Typically scales well with the number of features p
- Generally interpretable
- Can be used with any inducer

Negative

- Does not take specifics of the inducing algorithm into account
- Redundant features will have similar weights
- Often univariate