NOVA IMS Information Management School

Feature selection

Master in Data Science and Advanced Analytics
BA and DS majors

Roberto Henriques

Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Feature extraction, transformation and selection

- Feature Extraction and Engineering (extract something from them)
  - Texts(ngrams, word2vec, tf-idf, …)
  - Images(CNN'S, texts)
  - Geospatial data(lat, long)
  - Date and time(day, month, week, year, rolling based)
  - Time series
  - Dimensional Reduction Techniques (PCA, SVD, Eigen-Faces,…)
  - Clustering (DBSCAN,…)
- Feature transformations (transforming them to make sense)
  - Normalization and changing distribution(Scaling)
  - PCA
  - Filling in the missing values(median filling etc)
- Feature selection (building your model on these selected features)
  - Filter methods
  - Wrapper methods
  - Embedded methods

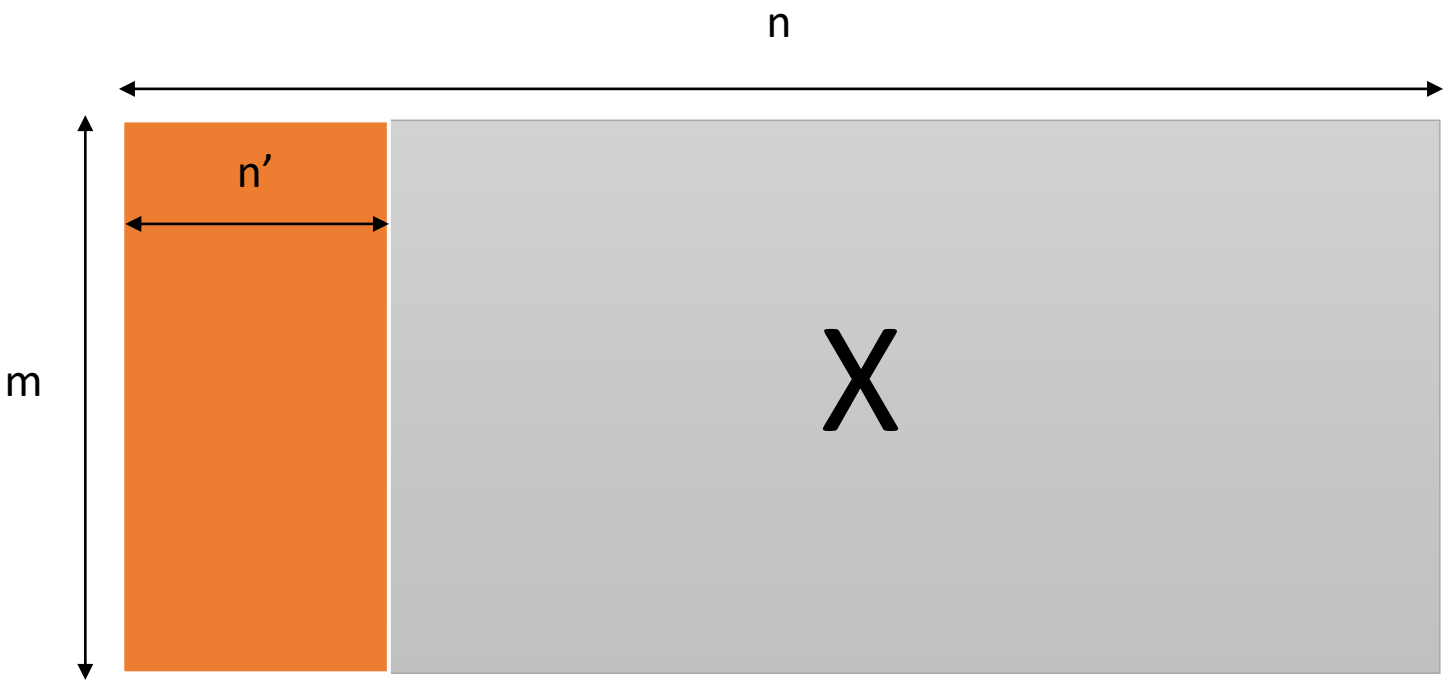
Feature selection

- Feature selection methods are intended to reduce the number of input variables to those that are believed to be most useful to a model in order to predict the target variable.

- **Why?**
  - Simple models are easier to interpret
  - Shorter training time
  - Enhanced generalization by reducing overfitting
  - Variable redundancy

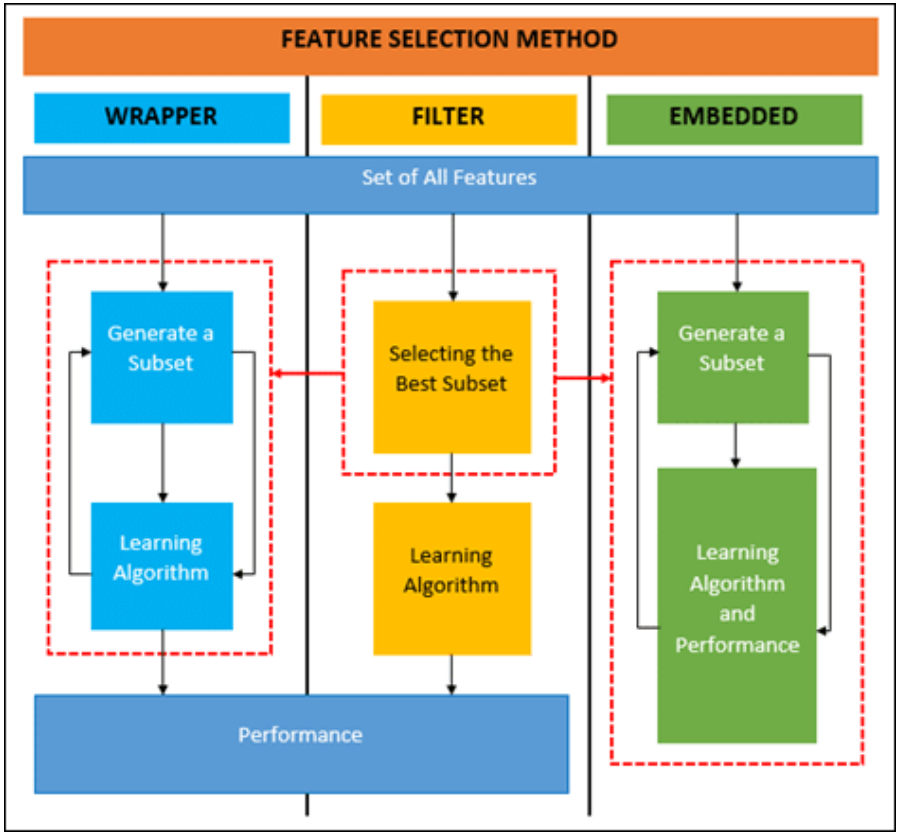
Feature Selection

- Thousands to millions of low level features
  - select the most relevant one to build better, faster, and easier to understand learning machines.



Feature Selection

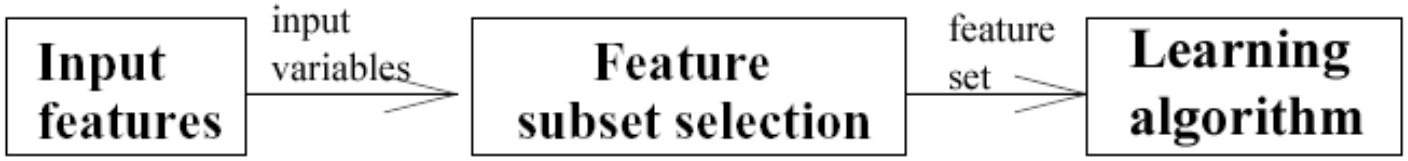
- Filter methods
  - ranks features or feature subsets independently of the predictor (classifier)
- Wrapper methods
  - uses a classifier to assess features or feature subsets
- Embedded methods
  - learn which features best contribute to the accuracy of the model while the model is being created



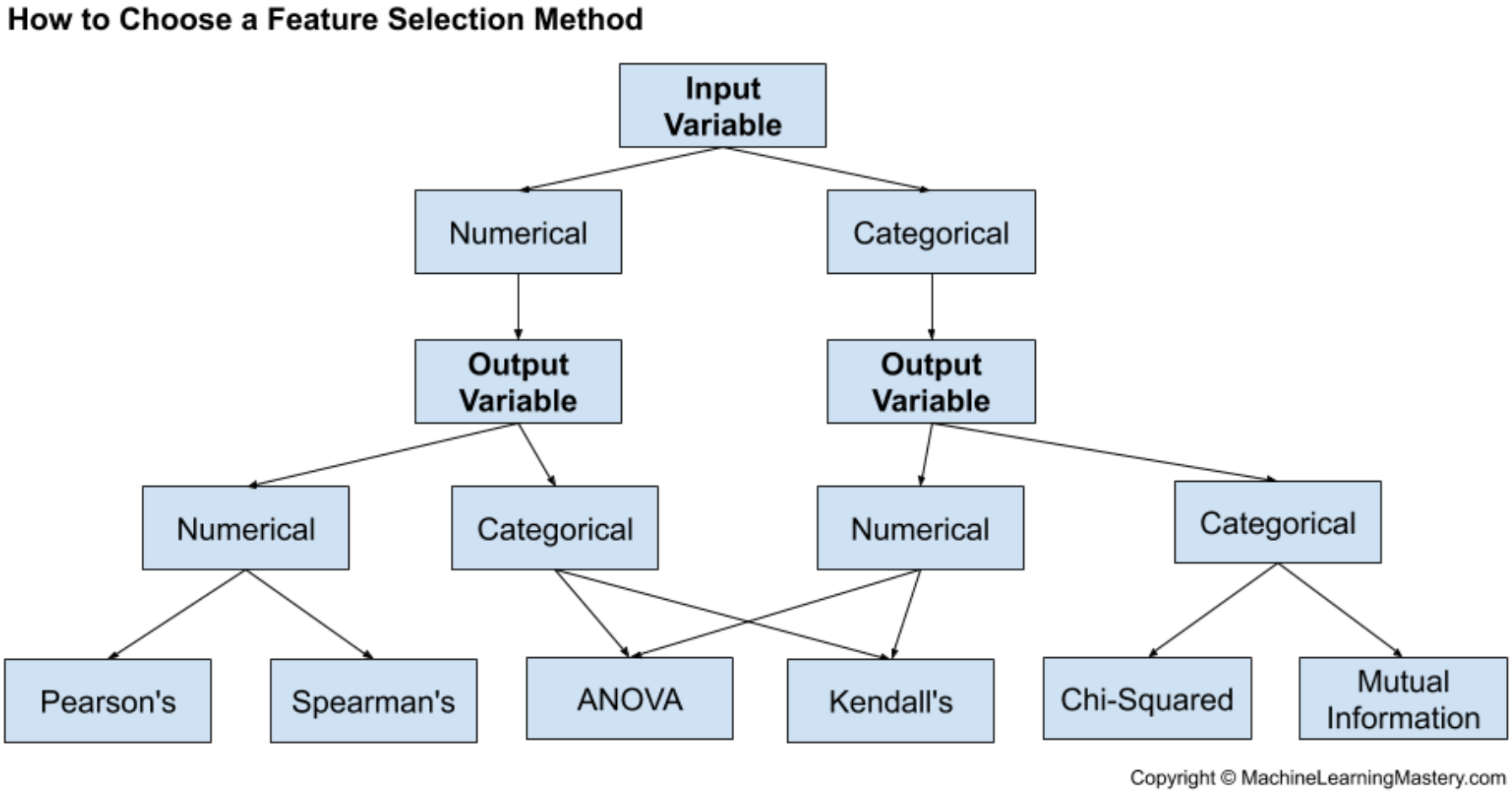
https://www.researchgate.net/publication/333168554/figure/fig2/AS:759537661407232@1558099298961/Work-Flow-for-Each-Feature-Selection-Method-Refer-Figure-2-feature-selection-methods.ppm

Filter methods

- use statistical techniques to evaluate the relationship between each input variable and the target variable
  - used filter those input variables to be used in the model
- statistical measures dependent upon the variable data types
- often univariate and consider the feature independently, or with regard to the dependent variable.
- Select subsets of variables independently of the used classifier



Which statistical tests to apply?



https://3qeqpr26caki16dnhd19sv6by6v-wpengine.netdna-ssl.com/wp-content/uploads/2019/11/How-to-Choose-Feature-Selection-Methods-For-Machine-Learning.png

- Pearson's correlation coefficient
  - measures linear correlation between two variables
- Spearman's rank coefficient
- ANOVA correlation coefficient (linear).
- Kendall's rank coefficient (nonlinear).
- Chi-Squared test (contingency tables).
- Mutual Information.

Pearson's correlation coefficient

- Pearson correlation coefficient

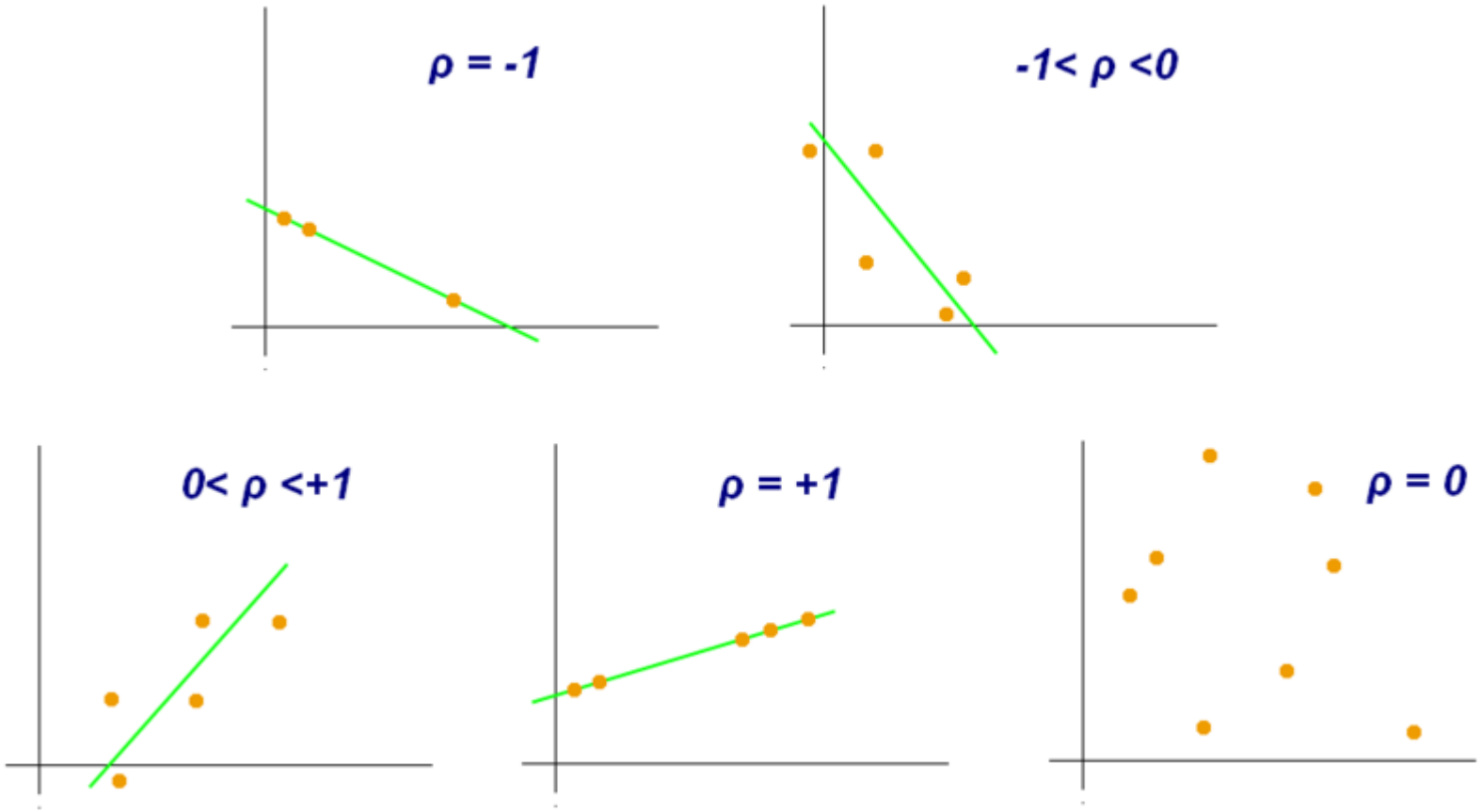
$$\mathrm{R}(f_i, y) = \frac{\mathrm{cov}(f_i, y)}{\sqrt{\mathrm{var}(f_i)\,\mathrm{var}(y)}}$$

- Estimate for m samples:

$$R(f_i, y) = \frac{\sum_{k=1}^{m}\left(f_{k,i} - \overline{f}_i\right)\left(y_k - \overline{y}\right)}{\sqrt{\sum_{k=1}^{m}\left(f_{k,i} - \overline{fi}\right)^2 \sum_{k=1}^{m}\left(y_k - \overline{y}\right)^2}}$$

- The higher the correlation between the feature and the target, the higher the score!

Pearson's correlation coefficient



https://upload.wikimedia.org/wikipedia/commons/thumb/3/34/Correlation_coefficient.png/400px-Correlation_coefficient.png

Spearman's rank correlation coefficient

We do care about the order (nonlinear correlation)

$$r_s = \rho_{\mathrm{rg}_X, \mathrm{rg}_Y} = \frac{\mathrm{cov}(\mathrm{rg}_X, \mathrm{rg}_Y)}{\sigma_{\mathrm{rg}_X} \sigma_{\mathrm{rg}_Y}},$$

- nonparametric measure of rank correlation (statistical dependence between the rankings of two variables).
- It assesses how well the relationship between two variables can be described using a monotonic function.
- The Spearman correlation between two variables is equal to the Pearson correlation between the rank values of those two variables

Variable ranking

- Can variables with small score be automatically discarded ?
- Can a small score variable be useful together with others ?
- Can two variables that are useless by themselves can be useful together?

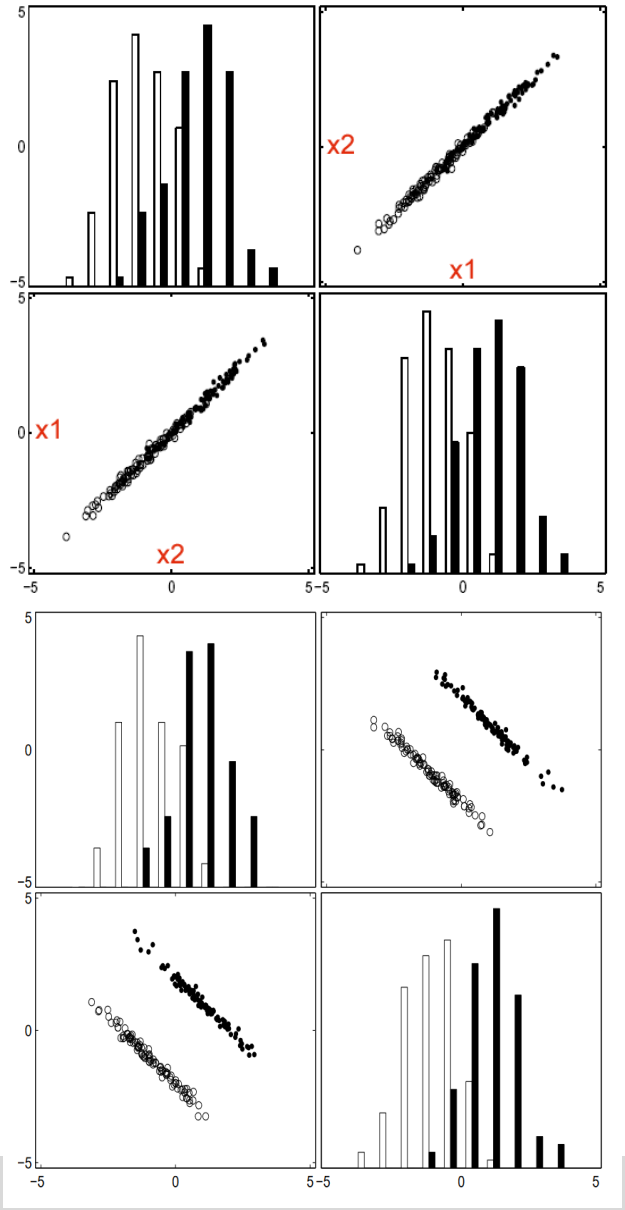
Can variables with small score be discarded?

- Example with high correlation between x_1 and x_2.

  No gain in separation ability by using two variables instead of just one!

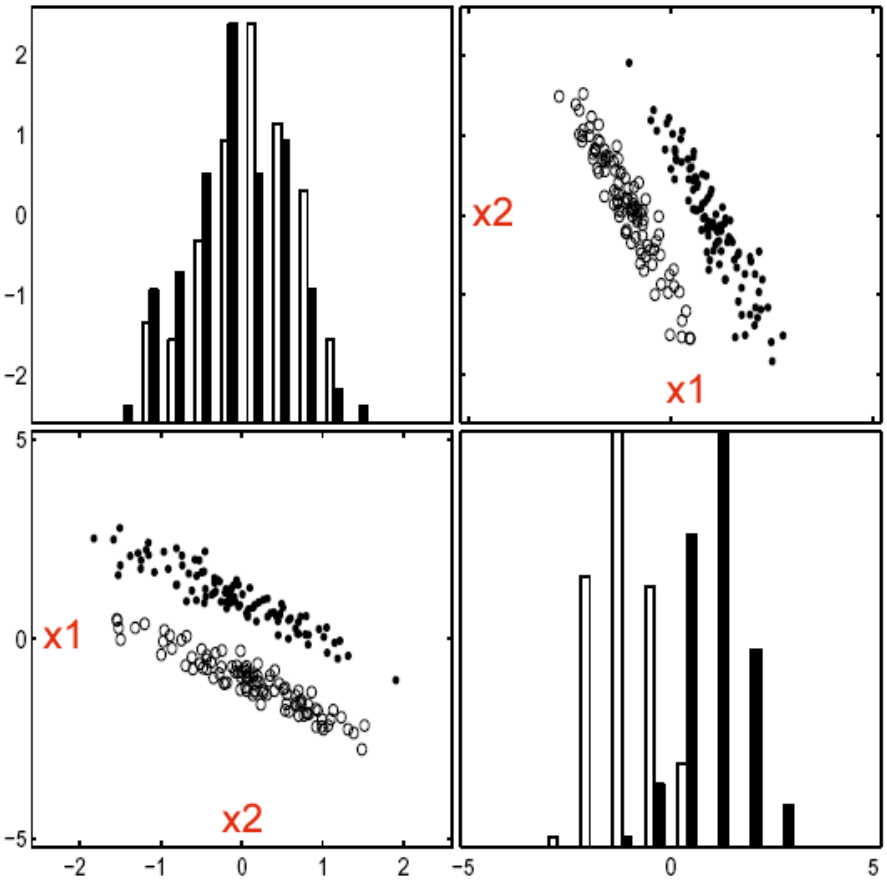
- Perfectly correlated variables are truly redundant in the sense that no additional information is gained by adding them

- Very high variable correlation does not mean the absence of variable complementarity.



Can variables with small score be discarded?

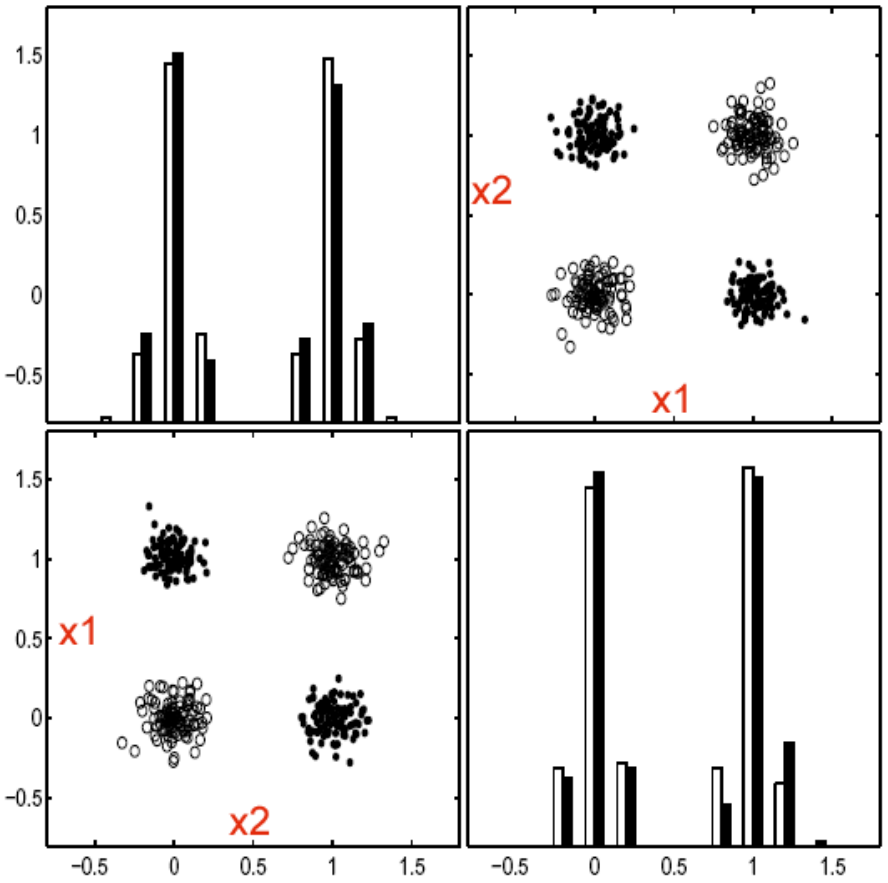
- Even variables with small score can improve class separability!
  - the two class conditional distributions have identical covariance matrices
- x_2 is "useless"
- Still, the two-dimensional separation is better than the separation using the x_1 alone.
- a completely useless variable by itself can provide a significant improvement when taken with others



https://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf

Can a useless variable be useful together with others?

- correlation between variables and target are not enough to assess relevance!
- correlation/covariance between pairs of variables has to be considered too!
- **diversity** of features



Information Theory

- Information Theoretic Criteria
- Most approaches use (empirical estimates of) **mutual information** between features and the target:

$$I(x_i, y) = \int_{x_i} \int_{y} p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)\, p(y)}\, dxdy$$

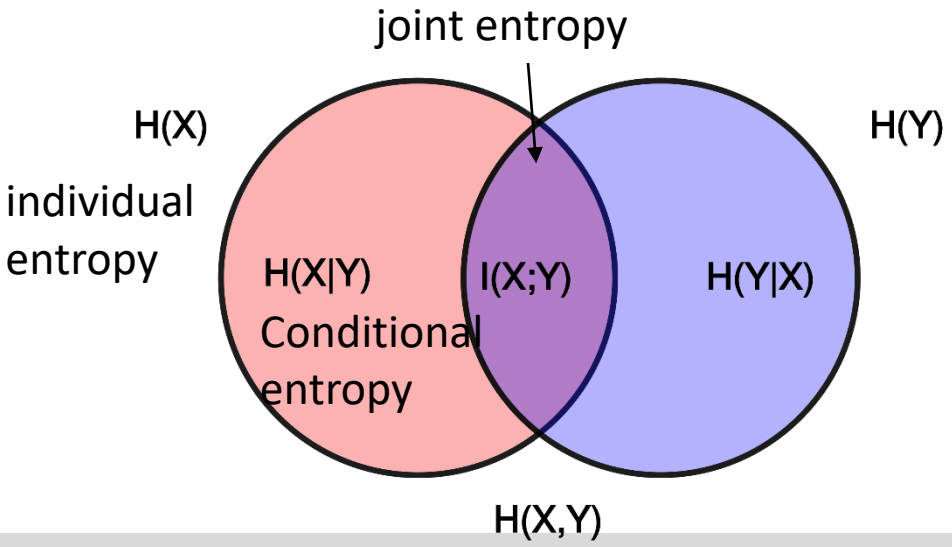
- Case of discrete variables:

$$I(x_i, y) = \sum_{x_i} \sum_{y} P(X = x_i, Y = y) \log \frac{P(X = x_i, Y = y)}{P(X = x_i) P(Y = y)}$$

(probabilities are estimated from frequency counts)

Mutual information

- Mutual information can also detect non-linear dependencies among variables!
- But harder to estimate than correlation!
- It is a measure for "how much information (in terms of entropy) two random variables share"



Mutual information

- inconvenient to use directly for feature ranking
  - it is not a metric and not normalized, so the MI values can be incomparable between two datasets.
  - inconvenient to compute for continuous variables
    - in general the variables need to be discretized by binning, but the mutual information score can be quite sensitive to bin selection

Maximal information coefficient

- Developed to address MI problems
- It searches for optimal binning and turns mutual information score into a metric that lies in range [0;1]

$$MIC(x, y) = \max\{I(x, y)/\log_2 \min\{n_x, n_y\}\}$$

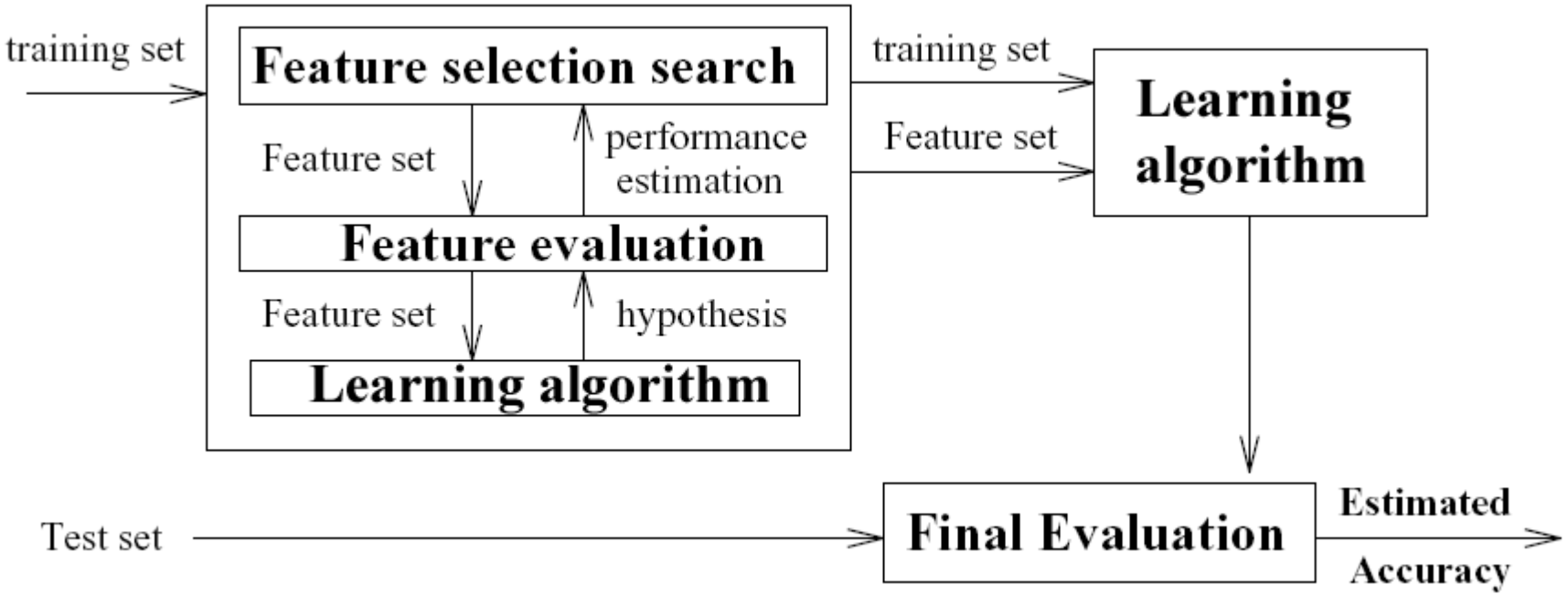
- **MIC(X,Y)** is the mutual information between random variables X and Y normalized by their minimum joint entropy.
- MIC is the percent of a variable Y that can be explained by a variable X

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3325791/

Wrapper methods

- consider the selection of a set of features as a search problem
  - different combinations are prepared, evaluated and compared to other combinations
  - A predictive model us used to evaluate a combination of features and assign a score based on model accuracy
- search process
  - methodical - such as a best-first search,
  - stochastic - such as a random hill-climbing algorithm,
  - use heuristics - forward and backward passes to add and remove features.
- Example: Recursive feature elimination algorithm

Wrapper methods



RFE

- First, the algorithm fits the model to all predictors.
- Each predictor is ranked using its importance to the model.
- Let S be a sequence of ordered numbers which are candidate values for the number of predictors to retain ($S_1 > S_2$, …).
- At each iteration of feature selection, the S_i top ranked predictors are retained, the model is refit and performance is assessed.
- The value of S_i with the best performance is determined and the top S_i predictors are used to fit the final model.
- The algorithm has an optional step (line 1.9) where the predictor rankings are recomputed on the model on the reduced feature set. Svetnik *et al* (2004) showed that, for random forest models, there was a decrease in performance when the rankings were re-computed at every step. However, in other cases when the initial rankings are not good (e.g. linear models with highly collinear predictors), re-calculation can slightly improve performance.

RFE

Algorithm 1: Recursive feature elimination

1.1 Tune/train the model on the training set using all predictors
1.2 Calculate model performance
1.3 Calculate variable importance or rankings
1.4 **for** *Each subset size* S_i, $i = 1 \ldots S$ **do**
1.5 Keep the S_i most important variables
1.6 [Optional] Pre–process the data
1.7 Tune/train the model on the training set using S_i predictors
1.8 Calculate model performance
1.9 [Optional] Recalculate the rankings for each predictor
1.10 **end**
1.11 Calculate the performance profile over the S_i
1.12 Determine the appropriate number of predictors
1.13 Use the model corresponding to the optimal S_i

RFE

- One potential issue - over-fitting to the predictor set such that the wrapper procedure could focus on nuances of the training data
  - In the case of a very large number of uninformative predictors and one predictor randomly correlated with the outcome;
  - RFE would give a good rank to this variable and the prediction error would be lowered
  - Using a different test/validation this predictor show it was uninformative
  - This is referred to as “selection bias” by Ambroise and McLachlan (2002)

- Training data is being used for three purposes:
  - predictor selection
  - model fitting
  - performance evaluation
- Unless the number of samples is large, especially in relation to the number of variables, one static training set may not be able to fulfill these needs

RFE with CV

- To get performance estimates that incorporate the variation due to feature selection, the previous algorithm should be encapsulated inside an outer layer of resampling (e.g. 10-fold cross-validation)

Algorithm 2: Recursive feature elimination incorporating resampling

```
for Each Resampling Iteration do
    Partition data into training and test/hold–back set via resampling
    Tune/train the model on the training set using all predictors
    Predict the held–back samples
    Calculate variable importance or rankings
    for Each subset size S_i, i = 1 ... S do
        Keep the S_i most important variables
        [Optional] Pre–process the data
        Tune/train the model on the training set using S_i predictors
        Predict the held–back samples
        [Optional] Recalculate the rankings for each predictor
    end
end
Calculate the performance profile over the S_i using the held–back samples
Determine the appropriate number of predictors
Estimate the final list of predictors to keep in the final model
Fit the final model based on the optimal S_i using the original training set
```

process to build the model is ≠ process to check the model

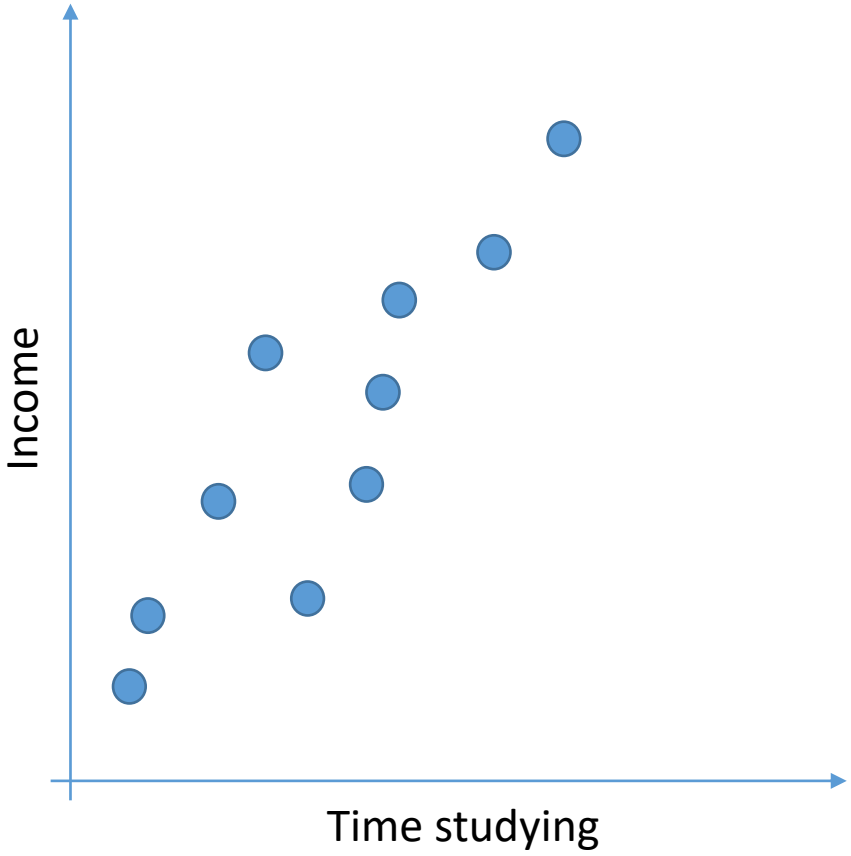
RFE with CV

- provide better estimates of performance but
- computationally burdensome
  - multiple processors, can parallelize line 2.1
- Multiple lists of the “best” predictors are generated at each iteration
  - provide a more probabilistic assessment of predictor importance than a ranking based on a single fixed data set.
  - At the end of the algorithm, a consensus ranking can be used to determine the best predictors to retain

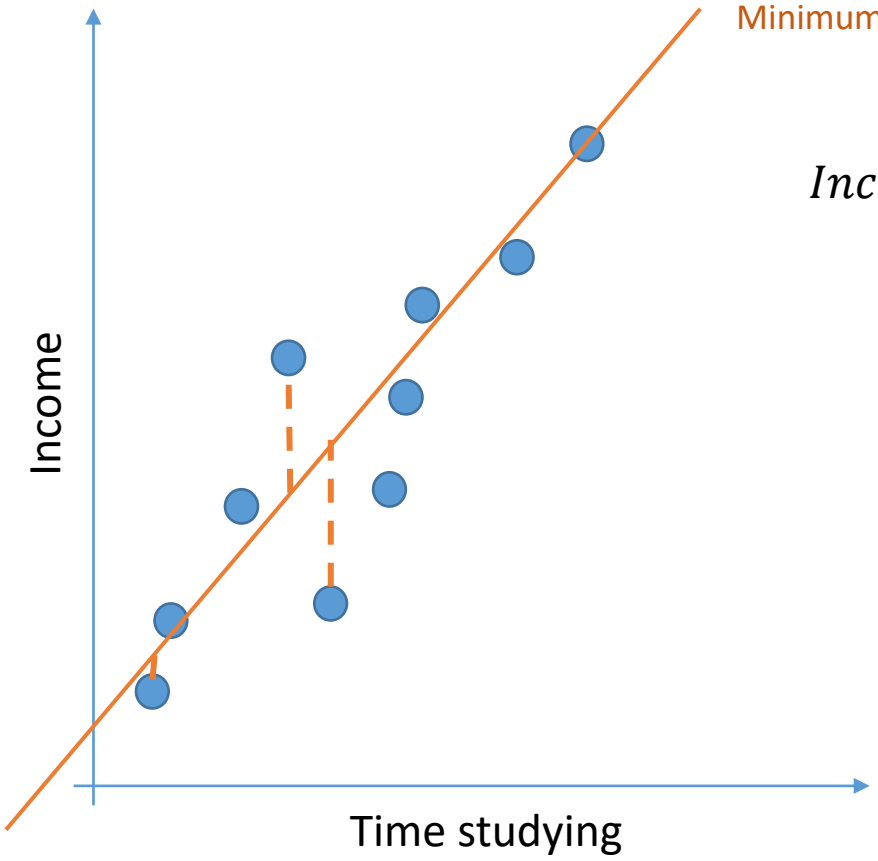
Embedded Methods

- learn which features best contribute to the accuracy of the model while the model is being created
- most common type are regularization methods
- Also called penalization methods
  - introduce additional constraints into the optimization of a predictive algorithm
  - bias the model toward lower complexity (fewer coefficients)
- Examples: LASSO, Elastic Net and Ridge Regression

The Bias-Variance Tradeoff



Ridge Regression



Least squares

Minimum some of squared residuals

$$Income = 0.8 + 1.1 \times TS$$

Ridge Regression

- *Ridge regression*
- The cost function

$$\sum_{i=1}^{M} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} w_j \times x_{ij} \right)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

- Minimizes

the sum of the squared residuals

+

λ x the slope²

Ridge Regression

- λ can range from 0 to positive infinity
- λ=0 then Ridge regression is same as Least Squares
- Increasing λ will promote a smaller slope (asymptotically to zero)
- How to choose λ
  - CV to determine the lowest variance
- Ridge Regression can solve for parameters when there is not enough data samples

LASSO regression

- The cost function for Lasso (Least Absolute Shrinkage and Selection operator) regression can be written as:

$$\sum_{i=1}^{M} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} w_j \times x_{ij} \right)^2 + \lambda \sum_{j=0}^{p} |w_j|$$

- Difference is instead of taking the square of the coefficients, magnitudes are taken into account.
- This type of regularization (L1) can lead to zero coefficients i.e. some of the features are completely neglected for the evaluation of output

NOVA IMS

Information Management School

Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa