Prof. Heitor Silvério Lopes
Prof. Thiago H. Silva

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

CPGEI

Data Mining & Knowledge Discovery

Class 6 – Feature Selection & Dimensionality Reduction

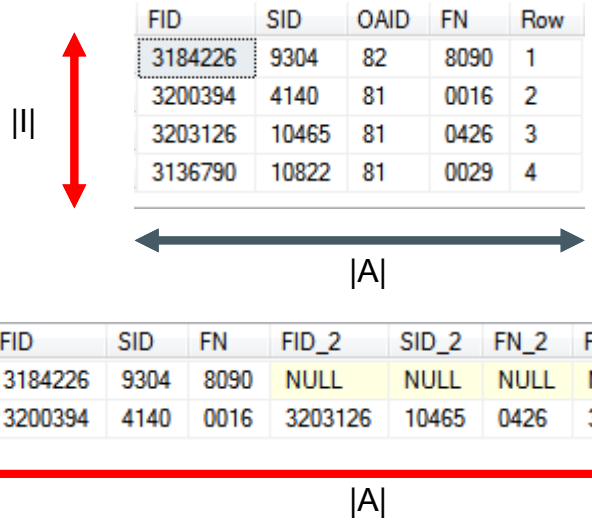
2025

Tasks x Methods in Data Mining

Tasks	Methods
Classification	Decision trees (C4.5), Classification rules, k-nearest-neighboors, Random forest, Support vector machine, Bayesian classifier, Neural network, Adaboost
Association Rules	Apriori, FP-growth, Eclat, Zigzag
Regression	Linear Regression, Polynomial regression, Logistic regression
Feature Selection & Dimensionality Reduction	Principal component analysis (PCA), Chi-square, Entropy, Information gain, t-SNE, Symmetric Uncertainty, Manifold learning
Clustering	K-means, Kohonen's self-organized map, Density-based scan, Hierarchical grouping
Data visualization *	Silhouette plot, scatter plot, heatmap, box plot, clusters, t-SNE

Shapes & dimensionality of datasets

- A dataset is a $|A| \times |I|$ matrix, where a $|A|$ is the number of Attributes and $|I|$ is the number of instances
  - Vertical datasets are |A|<<|I|
  - Horizontal datasets are |A|>>|I|
  - Usually, classification methods require a large amount of instances for training a model
  - A large number of atributes can difficult to find a suitable model, mainly those for descriptive analysis



What is the problem ?

- How to choose the best atributes of a given dataset for constructing a predictive or descriptive model ?
- If the number of attributes is small:
  - If the user has previous knowledge about the data domain, just select the more representative atributes
  - If the computational power is large, perform a factorial experimente, that is, **test all possible combinations** of attributes and compare results.
- If the number of attributes is (very) large:
  - It is imperative to use **a computationally efficient method** to select the most adequate attributes and discard the remaining

Further reasons for doing attribute selection

- Real-world datasets can have:
  - **Irrelevant attributes**: those which are not correlated with the target-atribute (class)
  - **Noisy attibutes**: those with variated degrees of noise due to the collection procedure
  - **Redundant attributes** : those strongly correlated with other atributes
- Using all attibutes can lead to poor predictive models
- Using less attibutes can:
  - Facilitate the visualization of multidimensional data
  - Create models more comprehensible for the user
  - Decrease training time of the algorithms
  - Increase the generalization capability of the training models

Feature selection X Dimensionality Reduction

- Both aim at decreasing the number of attibutes (=features) c
  dataset so that machine learning methods can be more efficient

- Attribute Selection:
  - Methods that select a **subset** of the original atributes by means of a specific quality criterion

- Dimensionality Reduction:
  - Methods that **create** other atributes by using the original ones

Attribute Selection modes

- Defining a dataset (DS) as a collection of ***n*** instances and ***m*** atributes (*n x m* matrix) an atribute selection method simply creates a mapping $\Phi=DS \rightarrow DS'$ such that $m' << m$
- Attibute selection can be performed in three diferente ways:
  - Selecting a subset of the original atributes, within ALL possible combinations (factorial experiment)
  - Ordering the atributes in decreasing order of a measure of suitability to the problem, and establishing a cuttof point
  - Weighting the atributes according to their suitability for a specific task (e.g. classification)

Groups of Methods for Feature Selection

- Filter methods:
  - Use statistical measures to measure a “quality” value for each atribute
  - Attributes are ordered according to this “quality” criterion
  - These methods are most commonly used for **unsupervised** problems
- Wrapper methods:
  - Use a search method do swap the possible space of attribute combinations
  - The quality of each set of selected attributes is evaluated by a predictive model (e.g. a classifier)
  - These methods are computationally expensive
  - Most commonly used for supervised problems

Filter methods for feature selection

- Information Theory-based methods:
  - Entropy (H)
  - Information Gain (IG)
  - Average Symmetric Uncertainty (ASU)
  - Minimum-redundance-Maximum-relevance (mRMR)
  - Relief
  - Fast Filter for Feature Selection (FCBF)
- Statistics-based methods:
  - Pearson correlation
  - Spearman correlation
  - Chi-Square Score (Mann-Whitley)
  - Z-Score (Signal-to-Noise Ratio)
  - Between and Within-Group Sum of Squares Ratio (BWSS)
  - Kruskall-Wallis (KW) score

Information Theory-based methods

- Entropy (H): it is a measure of uncertainty, dispersion or disorder, and varies between *0 ≤ H(x) ≤ 1*

$$H(X) = -\sum_{x \in X} p(x) \log(p(x))$$

- Information Gain (IG), Mutual Information (MI) or Kullback-Leibler Divergence (KLD): it is the information gain due to the choice of a specific attribute (or set of) with respect to the class

$$IG(X,Y) = H(X) + H(Y) - H(X,Y)$$

- Symmetric Mean Uncertainty (SU): it is derived from IG and corrects possible distortions of this method

$$SU(X,Y) = 2\frac{H(X) + H(Y) - H(X,Y)}{H(X) + H(Y)}$$

Fast Correlation-Based Filter (FCBF) algorithm

- FCBF selects atributes that are highly correlated with the target atribute (class) and have little (or no) correlation with the other variables. Steps:
  - The correlation measure used is the Symmetric Uncertainty (SU)
  - FCBF selects atributes correlated with the class above a given threshold
  - Then it detects the predominant correlations: when the correlation of an atribute A with the class is greater than the correlation of any other atribute with A
- FCBF was originally proposed for discrete attributes. For continuous attributes a discretization procedure must be applied prior to the use of the algorithm
- FCBF is efficient for high-dimensionality datasets

Yu, L. & Liu, H., Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. Proc. 20th Int. Conf. Machine Learning, 2003
https://jundongl.github.io/scikit-feature

Relief algorithm

- It is a “family” of filter methods for feature selection, inspired in Instance-Based Learning (IBL).
- Variants: Relief, ReliefF, RReliefF, I-Relief, tuned-Relief
- The algorithm is based on a user-defined relevance vector *R* and a threshold *L*
- Relief evaluates the differences between pairs of closest neighboring instances (in *n*-dimensional space):
  - If instances of the same class are close (regarding the threshold *L*) the *R* score decreases
  - If they are of diferente classes, the *R* score increases
- Each element of vector *R* corresponds to the relevance of an atribute for the class

Kira, K., Rendell, L.A. The feature selection problem: traditional methods and a new algorithm. Proc. AAAI conference, 1992

https://medium.com/@yashdagli98/feature-selection-using-relief-algorithms-with-python-example-3c2006e18f83

Comparison between FCBF and Relief

	FCBF	Relief
Metric	• Correlation	• Neighborhood
Computational complexity	***O**(instances^2 x atributes)*	***O**(instances^2 x atributes x iterations)*
Data type	• Discrete or discretized data	• Nominal or numeric
Advantages	• Good for high-dimensional data • Considers both redundance and relevance	• Efficient when there are iteractions between atributes • Robust to noise
Disadvantages	• Based on linear correlation, may not capture nonlinear correlations	• Do not remove redundant atributes • Performance dependent on the choice of #neighbors

Wrapper methods for feature selection

- They are computationally expensive because they sequentially evaluate a large number of attribute subsets.
- Search models:
  - Random Search, Exhaustive Search (Grid Search),
  - Genetic algorithm,
  - Swarm methods: PSO, ACO, ABC..
  - Classical metaheuristics:Tabu Search, simulated annealing, scatter Search
- Classification methods:
  - OneR, Decision trees (C4.5), Neural network, SVM, KNN
- Quality metrics with cross-validation:
  - Accuracy (for dataset with balanced classes), F1, AUC

Case study #1: Pima indians diabetes – Filter methods

- Instances: 768 (500 neg., 268 pos.)
- Attibutes: **8** (Number of pregnances, Plasma glucose concentration, Diastolic blood pressure, Skin tickness, 3-hour serum insulin, Body-mass index, Diabetes pedigree function, Age)
- Classes: Positive, Negative

	#	Info. gain	Gain ratio	χ^2	ReliefF	FCBF
1 N Glucose		0.170	0.085	139.901	0.019	0.131
2 N Age		0.081	0.041	62.029	0.005	0.059
3 N BMI		0.079	0.039	53.744	0.010	0.057
4 N DiabetesPedigreeFunction		0.022	0.011	16.143	0.005	0.015
5 N Insulin		0.055	0.030	8.780	0.009	0.000
6 N Pregnancies		0.043	0.021	34.316	0.014	0.000
7 N SkinThickness		0.036	0.018	5.262	0.010	0.000
8 N BloodPressure		0.015	0.007	12.918	0.012	0.000

Top-1 →
Top-3 →

	Feature	Info. gain	Gain ratio	χ^2	ReliefF	FCBF	average
2	Glucose	1.000	1.00	1	1.00	1.000	0.8
6	BMI	0.41088	0.4108	0.360092	0.4562	0.43144	0.341876
8	Age	0.42774	0.4284	0.421625	0.0783	0.44712	0.276309
1	Pregnancies	0.17826	0.1807	0.215796	0.7983	0.00002	0.231446
3	BloodPressure	0.000	0.00	0.0568603	0.8275	0.000	0.165493
5	Insulin	0.25635	0.2926	0.0261313	0.0391	0.00002	0.117607
7	DiabetesPedigr...	0.04639	0.0463	0.0808189	0.3687	0.11583	0.11544
4	SkinThickness	0.13884	0.1398	0	0.00	0.00001	0.0557263

**Average of Info.Gain, Gain Ratio, ReliefF, FCBF

Case study #1: Pima indians diabetes – Filter methods

- Classification using decision tree and neural network
- *10-fold cross-validation*
- Evaluation metric: F1 (unbalanced classes) and tree complexity

Features	Nodes/leaves	Classification Results
Top-1 (OneRule)	103 / 52	Tree (1): AUC 0.741, CA 0.715, F1 0.698, Prec 0.703, Recall 0.715, MCC 0.332 Neural Network: AUC 0.785, CA 0.745, F1 0.731, Prec 0.738, Recall 0.745, MCC 0.408
Top-3	91 / 46	Tree (2): AUC 0.793, CA 0.762, F1 0.759, Prec 0.758, Recall 0.762, MCC 0.465 Neural Network: AUC 0.829, CA 0.768, F1 0.762, Prec 0.763, Recall 0.768, MCC 0.473
All features	91 / 46	Tree: AUC 0.748, CA 0.733, F1 0.730, Prec 0.728, Recall 0.733, MCC 0.400 Neural Network: AUC 0.824, CA 0.758, F1 0.755, Prec 0.754, Recall 0.758, MCC 0.458

All atributes → worse result !

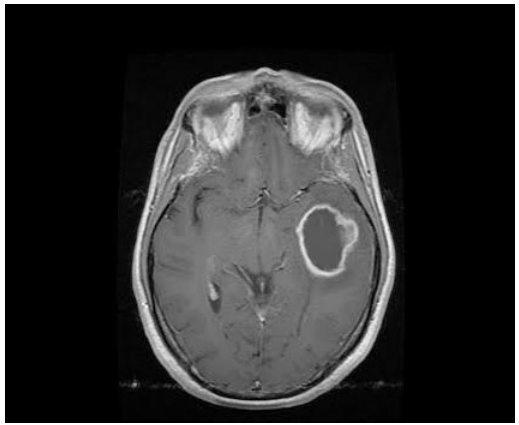
Case study #1: Pima indians diabetes – Wrapper methods

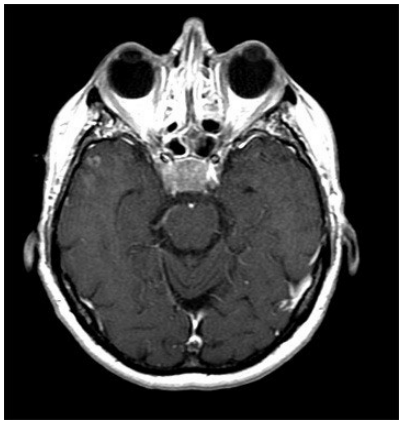
- Orange does **not** have Wrapper methods for feature selection !
- Classifier + Search method (Weka)

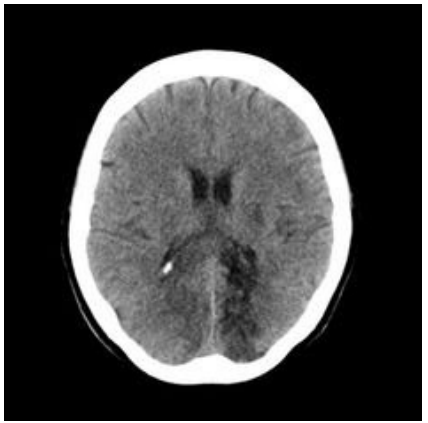
Attributes	OneRule	JRip	JRip	J48	Count
	Random	Random	Genetic Algorithm	Exhaustive	
#Pregnancies			✓		1
Glucose	✓	✓	✓	✓	4
Blood Pressure	✓	✓		✓	3
Skin Tickness	✓		✓		2
Insulin			✓		1
BMI	✓	✓	✓	✓	4
Diabetes Pedigree					0
Age		✓	✓	✓	3

Case study #2: Brain tumor images classificaton

- Magnetic resonance images of three classes of tumors (glioma, meningioma, pituitary) plus a control group (no-tumor)
- Data available at: https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri
- Feature extractor: Squeezenet NN (**1000** numeric features)
- Train / test: 2870/364 images







Case study #2: Brain tumor images classificaton

- All runs use 5x stratified cross-validation in the training dataset

All1000 features

Model	AUC	CA	F1	Prec	Recall	MCC
Tree Baseline-All	0.771	0.686	0.685	0.684	0.686	0.571
Neural Network	0.983	0.903	0.903	0.903	0.903	0.867

Top-1 Information Gain

Model	AUC	CA	F1	Prec	Recall	MCC
Tree Baseline Top-1 IG	0.592	0.362	0.362	0.369	0.362	0.127
Neural Network	0.675	0.434	0.402	0.442	0.434	0.216

Top-23 FCBF

Model	AUC	CA	F1	Prec	Recall	MCC
Tree-23 FCBF	0.785	0.684	0.685	0.687	0.684	0.568
Neural Network	0.960	0.831	0.831	0.831	0.831	0.769

Top200 Information Gain

Model	AUC	CA	F1	Prec	Recall	MCC
Tree Top-200-IG	0.781	0.693	0.692	0.691	0.693	0.580
Neural Network	0.979	0.886	0.886	0.886	0.886	0.845

Top-100 Relief

Model	AUC	CA	F1	Prec	Recall	MCC
Tree Top-100 Relief	0.777	0.685	0.685	0.685	0.685	0.570
Neural Network	0.975	0.873	0.873	0.873	0.873	0.826

Top200 Information Gain Ratio

Model	AUC	CA	F1	Prec	Recall	MCC
Tree Top-200 IGR	0.780	0.692	0.691	0.690	0.692	0.579
Neural Network	0.979	0.892	0.892	0.892	0.892	0.853

Dimensionality Reduction

- It is **not** an attribute selection method, but a transformation of the original attributes into new and, hopefully, more discrimatory ones
- Main methods:
  - **Principal Component Analysis (PCA)**
  - Singular Vector Decomposition (SVD)
  - Independent Component Analysis (ICA)
  - Locally Linear Embedding (LLE)
  - Isometric Feature Mapping (ISOMAP)
  - Non-negative Matrix Factorization (NMF)
  - Autoencoders
  - Some of these methods are available in the Pyhton's scikit-learn library:

https://machinelearningmastery.com/dimensionality-reduction-algorithms-with-python/

Principal Components Analysis (PCA)

- Also known as Karhunen-Loève transform, but invented by Pearson (1901)
- It is an orthogonal linear transformation method that aims to find new linearly uncorrelated atributes.
- Besides Dimensionality Reduction, the method is also used for Exploratory Data Analysis tasks
- Considering *D* as the number of original atributes, PCA's computational complexity is given by:
  - Time: $O(D^3)$,
  - Memory: $O(D^2)$

Principal Components Analysis (PCA)

- Each Principal Component (PC) represents a small amount of the **variance** in the data
- The first PC captures the largest possible variance in the data
- Each subsequent component has the maximum **remaining variance ***, considering the constraint of being **orthogonal** (uncorrelated) to all the previous ones
- Usually, a number of componentes is chosen such that 90-95% of the original variance is explained (when possible)

* **Variance**: it is a measure of statistical dispersion, representing how far its values are from the expected value

Principal Components Analysis (PCA)

- Variance of a single variable: $$\mathrm{Var}(X) = \frac{1}{n}\sum_j (\bar{x} - x_j)^2 = \sigma_X^2$$
- Covariance between two variables: $$\mathrm{Cov}(X, Y) = \frac{1}{n}\sum_j (\bar{x} - x_j)(\bar{y} - y_j) = \sigma_{XY}$$
- Covariance matrix for **n** variables: $$\mathrm{C} = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22}^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn}^2 \end{pmatrix}$$

https://wilkelab.org/SDS375/slides/dimension-reduction-1.html

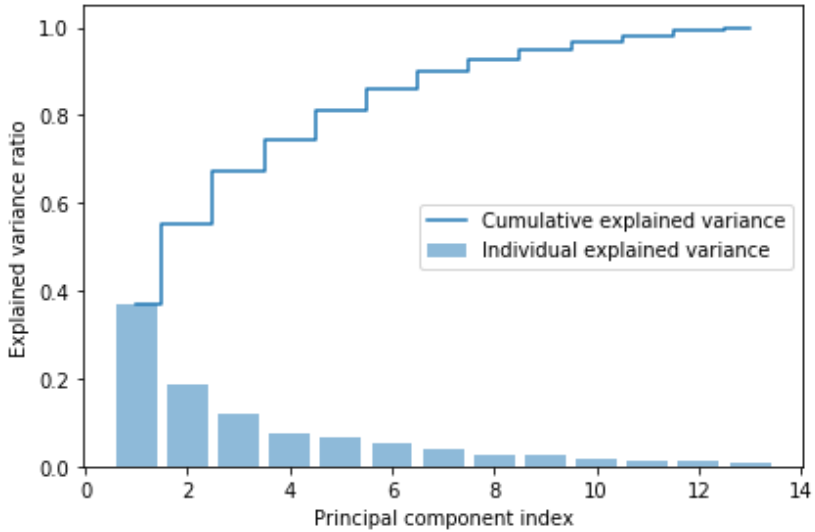
Principal Components Analysis (PCA)

- Diagonalization of the covariance matrix: $\mathrm{C} = \mathrm{U}\mathrm{D}\mathrm{U}^{\mathrm{T}}$
- Where:
  - U = rotation matrix, D = diagonal matrix
  - λ_j^2 = the variance explained by the *j* component (or eigenvalues)
  - The componentes are uncorrelated
  - The covariance between the componentes is zero

$$= \mathrm{U} \begin{pmatrix} \lambda_1^2 & 0 & \dots & 0 \\ 0 & \lambda_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n^2 \end{pmatrix} \mathrm{U}^{\mathrm{T}}$$

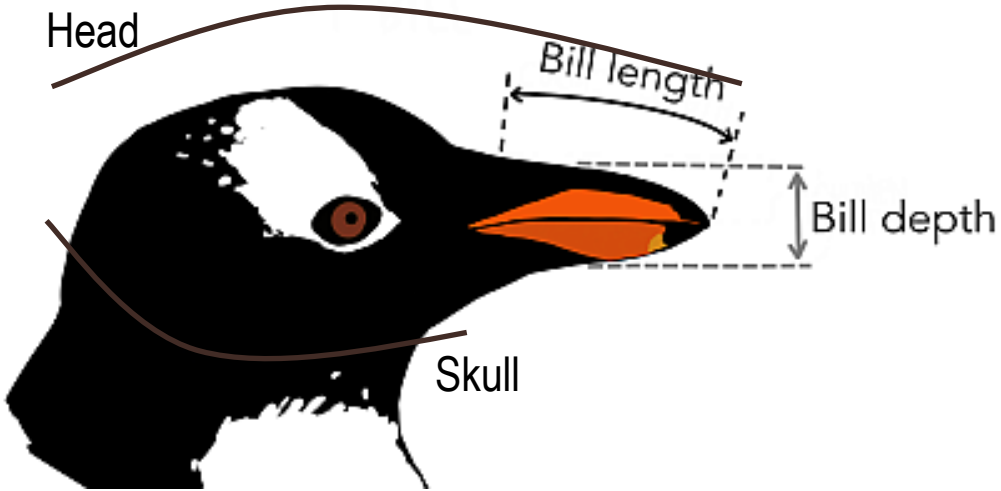
Principal Components Analysis (PCA)

- The first componente (PC1) accounts for the dimension where there is the greatest variance in the data
- The greater the number of PCA compoenents, the greater the **cumulative variance** “explained”



Case study #3: Bluejay dataset

- IMPORTANT: this is a **trivial** problem, since there are only 6 variables
- Variables (6): *BillDepth, BillWidth, BillLenght, Head, Mass, Skull*
- Meta-attribute: *Sex* (0=*female*, 1=*male*)



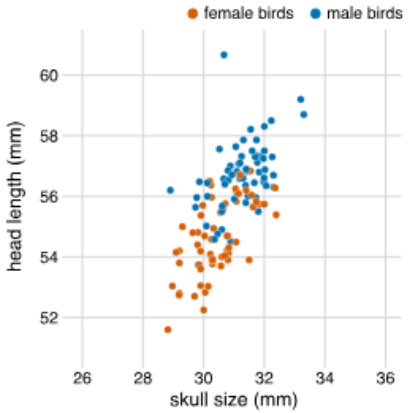
Case study #3: Bluejay dataset

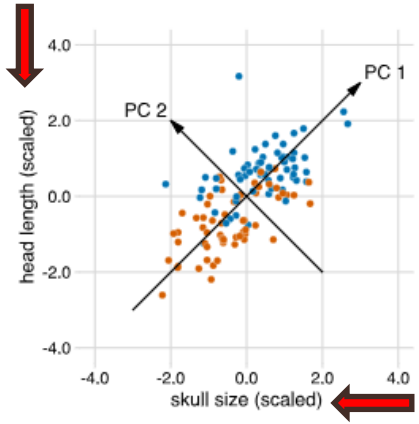
- Numer of instances: 124

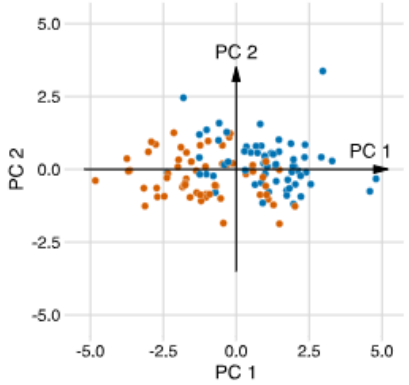
bird_id	sex	bill_depth_mm	bill_width_mm	bill_length_mm	head_length_mm	body_mass_g	skull_size_mm
0000-00000	M	8.26	9.21	25.92	56.58	73.30	30.66
1142-05901	M	8.54	8.76	24.99	56.36	75.10	31.38
1142-05905	M	8.39	8.78	26.07	57.32	70.25	31.25
1142-05907	F	7.78	9.30	23.48	53.77	65.50	30.29
1142-05909	M	8.71	9.84	25.47	57.32	74.90	31.85
1142-05911	F	7.28	9.30	22.25	52.25	63.90	30.00
1142-05912	M	8.74	9.28	25.35	57.12	75.10	31.77
1142-05914	M	8.72	9.94	30.00	60.67	78.10	30.67
1142-05917	F	8.20	9.01	22.78	52.83	64.00	30.05
1142-05920	F	7.67	9.31	24.61	54.94	67.33	30.33
1142-05930	M	8.78	8.83	25.72	56.54	76.40	30.82

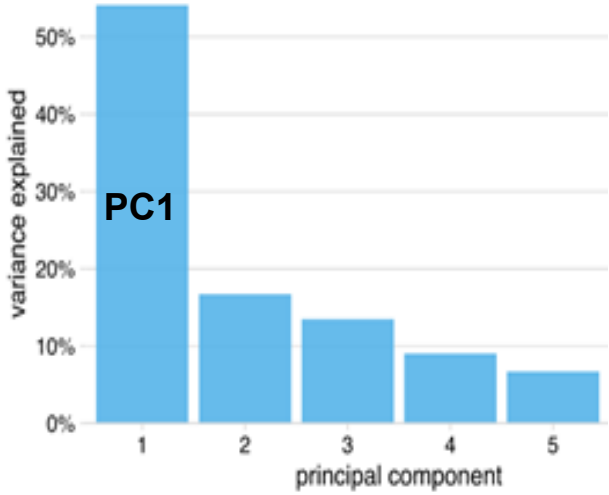
Case study #3: Bluejay dataset

- PCA aligns the major axes with directions of maximum variation in the data
- Using PCA, the 6-dimension original space is represented by its principal components with the highest variance









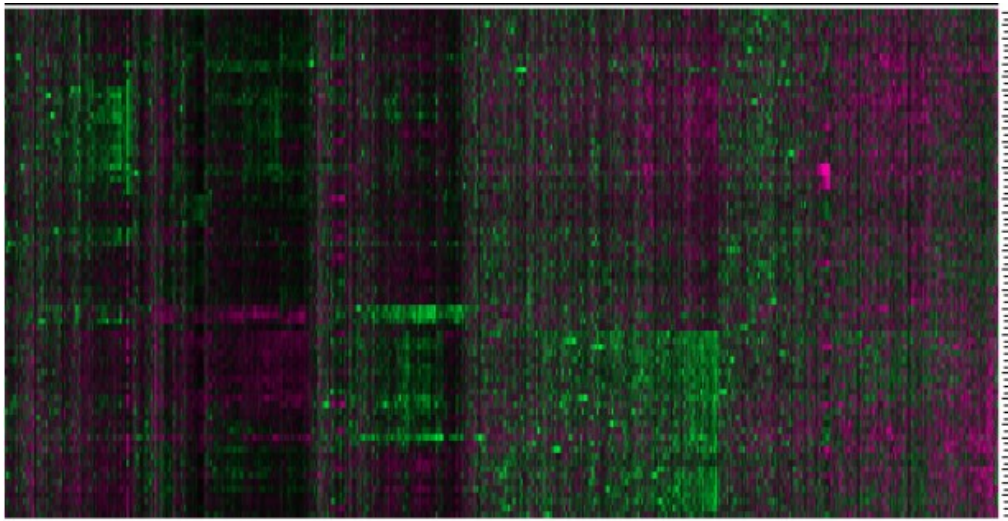
Case study #4: Smoking effects on B lymphocytes dataset

- The B lymphocytes attack invaders outside the cells
- They are responsible for adaptive immunity and are associated with the onset and development of many diseases induced by continuous cigarette use.
- The analysis of gene expression of these cells may provide useful information about the relationship between the cells and diseases

Case study #4: Smoking effects on B lymphocytes dataset

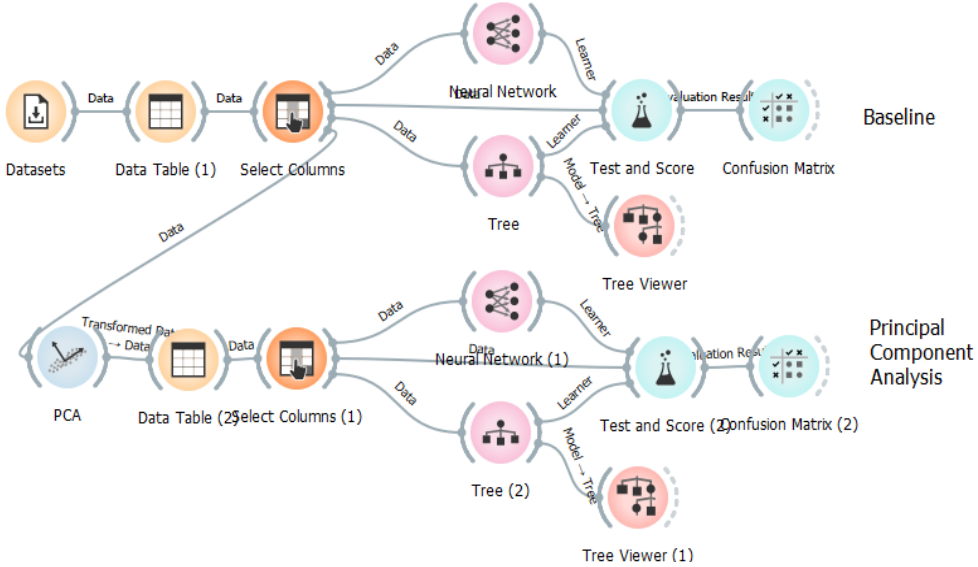
- Gene expression data from peripheral blood B cells of 39 smokers and 40 non-smokers, all female
- The data are from a sample of 3000 genes out of a total of 14,500 genes evaluated

Pan F. et al. Impact of female cigarette smoking on circulating B cells in vivo: the suppressed ICOSLG, TCF3, and VCAM1 gene functional network may inhibit normal cell function. **Immunogenetics**, 62(4), 237-251, 2010. https://www.ncbi.nlm.nih.gov/sites/GDSbrowser/?acc=GDS3713



Case study #4: Smoking effects on B lymphocytes dataset

Dimensionality reduction using PCA



All features

Predicted

Actual	cigarette smoke	control	Σ
cigarette smoke	30	9	39
control	11	29	40
Σ	41	38	79

Best PCA with 10 PC

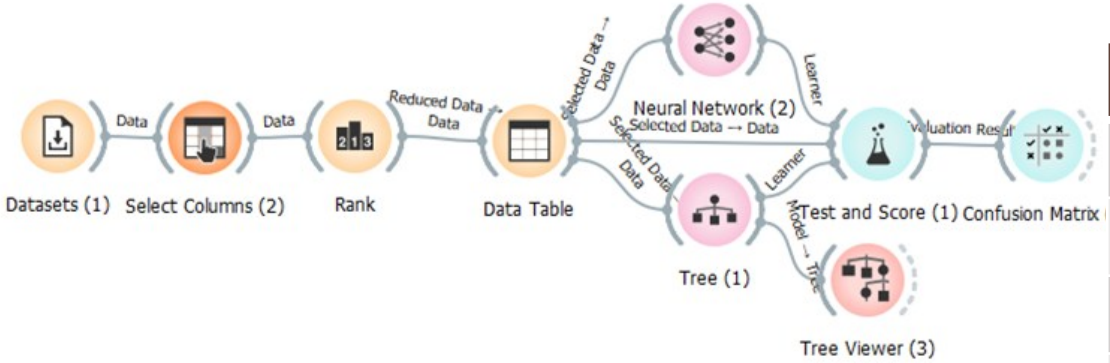
Predicted

Actual	cigarette smoke	control	Σ
cigarette smoke	35	4	39
control	5	35	40
Σ	40	39	79

# PC's	%variance	F1 DT	F1 NN
No PC		0,747	**0,911**
1 PC	8%	**0,582**	**0,620**
2 PC	14%	0,822	0,848
10 PC	34%	**0,835**	**0,899**
20 PC	49%	0,835	0,873
30 PC	61%	0,848	0,835
50 PC	80%	**0,835**	**0,543**
79 PC	99%	**0,835**	**0,187**

Case study #4: Smoking effects on B lymphocytes dataset

Feature selection



- Depending upon the classifier algorithm, the number of attributes (variables) can influence positivelly or negativelly in the classification result
- Comparison between: Relief, FCBF, InfoGain, and X^2

	#Attrib.	F1 DT	F1 NN
baseline	all	0,747	**0,911**
Relief	50	**0,810**	**0,975**
Relief	100	0,733	0,987
FCBF	25	**0,706**	**1,000**
FCBF	100	0,734	0,949
InfoGain	25/100	0,734	**0,911**
X^2	25	0,747	0,925
X^2	100	0,732	0,937

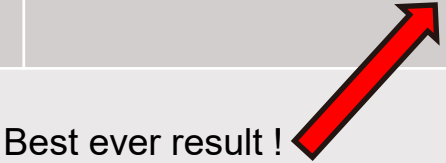
Case study #2 (revisited): Brain tumor images classification

- Using PCA for creating up to 90 new features from the original 1000, but using only 2

Method	(previous best result) Top-200 features (selected by Information Gain)	2 PCs
Results in the test set		

Model	AUC	CA	F1	Prec	Recall	MCC
Tree Top-200 IGR	0.780	0.692	0.691	0.690	0.692	0.579
Neural Network	0.979	0.892	0.892	0.892	0.892	0.853

Model	AUC	CA	F1	Prec	Recall	MCC
Tree 30PC	0.771	0.688	0.687	0.686	0.688	0.574
Neural Network	0.983	0.907	0.908	0.908	0.907	0.874



Additional links for further study

- Feature selection in Python using Relief for feature selection

  https://medium.com/@yashdagli98/feature-selection-using-relief-algorithms-with-python-example-3c2006e18f83

- Feature selection in Python using FCBF for feature selection:
  https://github.com/shiralkarprashant/FCBF

- Feature selection methods using Scikit-learn
  https://scikit-learn.org/stable/modules/feature_selection.html

- A comprehensive guide on Feature Selection (Kaggle):
  https://www.kaggle.com/code/prashant111/comprehensive-guide-on-feature-selection