Data Preparation & Building Numerical Predictors

KNIME AG

Data Preparation

Motivation

- **Real world data is “dirty“**
  - → Contains missing values, noises, outliers, inconsistencies
- **Comes from different information sources**
  - → Different attribute names, values expressed differently, related tuples
- **Different value ranges and hierarchies**
  - → One attribute range may overpower another
- **Huge amount of data**
  - → Makes analyis difficult and time consuming

Data Preparation

- Data Cleaning & Standardization (domain dependent)
- Aggregations (often domain dependent)
- Normalization
- Dimensionality Reduction
- Outlier Detection
- Missing Value Imputation
- Feature Selection
- Feature Engineering
- Sampling
- Integration of multiple Data Sources

Data Preparation: Normalization

Normalization: Motivation

Example:

- Lengths in cm (100 – 200) and weights in kilogram (30 – 150) fall both in approximately the same scale
- What about lengths in m (1-2) and weights also in gram (30000 – 150000)?
  → The weight values in mg dominate over the length values for the similarity of records!

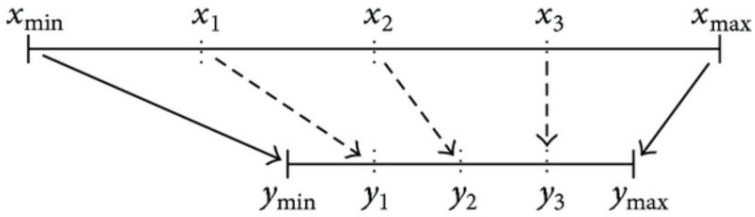
Goal of normalization:

- Transformation of attributes to make record ranges comparable

Normalization: Techniques

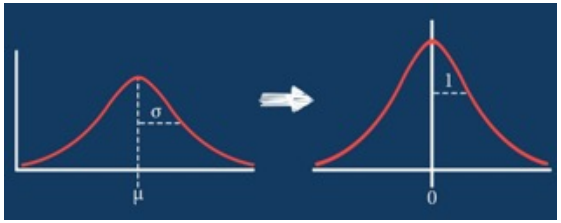
- **min-max normalization**

$$y = \frac{x - x_{min}}{x_{max} - x_{min}} (y_{max} - y_{min}) + y_{min}$$



- **z-score normalization**

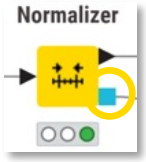
$$y = \frac{x - mean(x)}{stddev(x)}$$



- **normalization by decimal scaling**

$$y = \frac{x}{10^j}$$ where j is the smallest integer for $\max(y) < 1$

Here $[ymin, ymax]$ is $[-1,1]$



Data Preparation: Outlier Detection

Outlier Detection

- An outlier could be, for example, rare behavior, system defect, measurement error, or reaction to an unexpected event

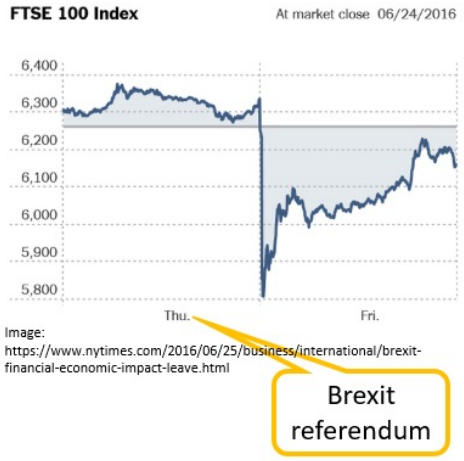
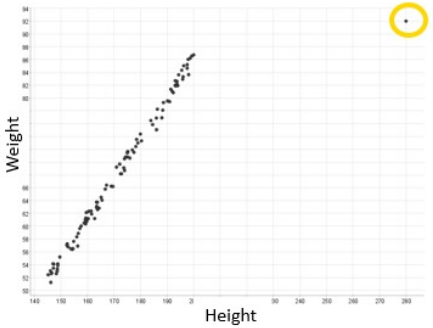


Image: https://www.nytimes.com/2016/06/25/business/international/brexit-financial-economic-impact-leave.html

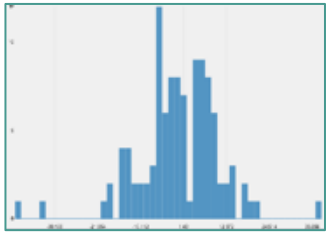


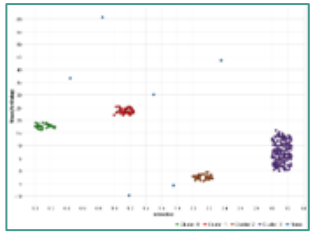
Outlier Detection: Motivation

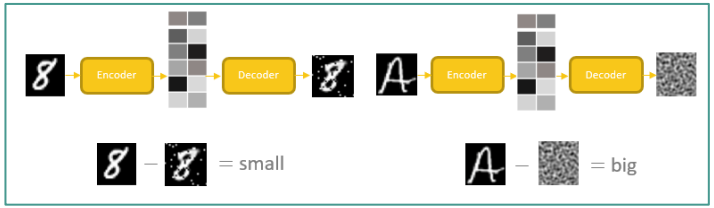
- Why finding outliers is important?
  - Summarize data by statistics that represent the majority of the data
  - Train a model that generalizes to new data
  - Finding the outliers can also be the focus of the analysis and not only data cleaning

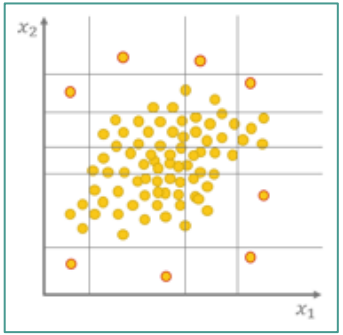
Outlier Detection Techniques

- Knowledge-based
- Statistics-based
  - Distance from the median
  - Position in the distribution tails
  - Distance to the closest cluster center
  - Error produced by an autoencoder
  - Number of random splits to isolate a data point
    from other data

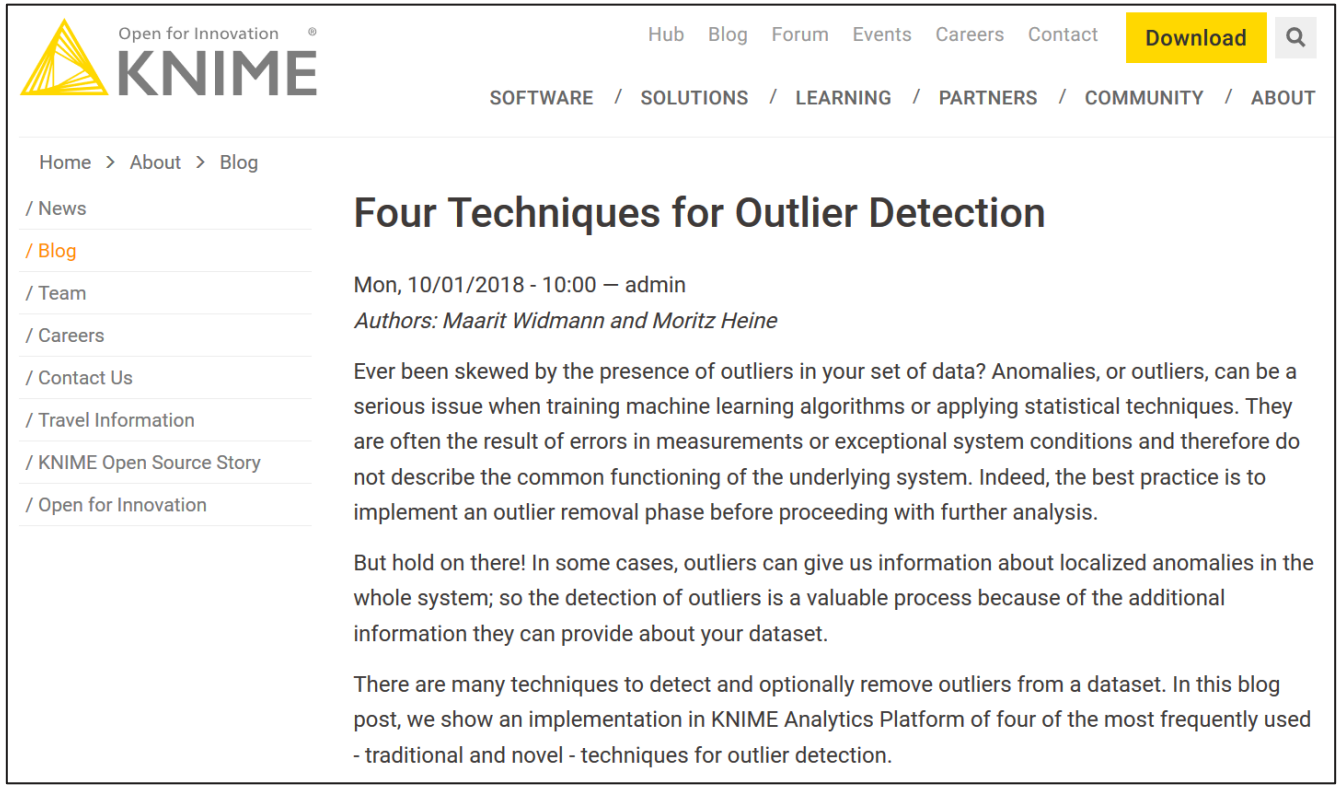








Material



https://www.knime.com/blog/four-techniques-for-outlier-detection

Data Preparation: Dimensionality Reduction

Is there such a thing as “too much data”?

“Too much data”:

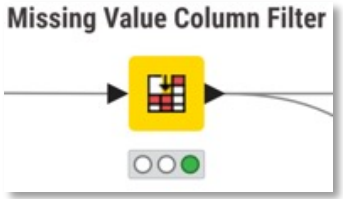
- Consumes storage space
- Eats up processing time
- Is difficult to visualize
- Inhibits ML algorithm performance
- Beware of the model: Garbage in → Garbage out

Dimensionality Reduction Techniques

- Measure based
  - Ratio of missing values
  - Low variance
  - High Correlation
- Transformation based
  - Principal Component Analysis (PCA)
  - Linear Discriminant Analysis (LDA)
  - t-SNE
- Machine Learning based
  - Random Forest of shallow trees
  - Neural auto-encoder

Missing Values Ratio

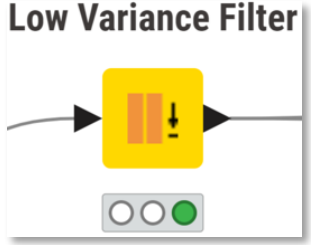
#	Row...	pclass Number (integer)	survived Number (integer)	name String	sex String	age Number (double)	sibsp Number (integer)	parch Number (integer)	ticket String	fare Number (double)	cabin String	embarked String	boat String	body Number (integer)	home.dest String
29	Row28	1	1	Bissette, M...	female	35	0	0	PC 17760	135.633	C99	S	8	?	?
30	Row29	1	1	Bjornstrom...	male	28	0	0	110564	26.55	C52	S	D	?	Stockholm, Sweden / Washington, DC
31	Row30	1	0	Blackwell, ...	male	45	0	0	113784	35.5	T	S	?	?	Trenton, NJ
32	Row31	1	1	Blank, Mr. ...	male	40	0	0	112277	31	A31	C	7	?	Glen Ridge, NJ
33	Row32	1	1	Bonnell, Mi...	female	30	0	0	36928	164.867	C7	S	8	?	Youngstown, OH
34	Row33	1	1	Bonnell, Mi...	female	58	0	0	113783	26.55	C103	S	8	?	Birkdale, England Cleveland, Ohio
35	Row34	1	0	Borebank, ...	male	42	0	0	110489	26.55	D22	S	?	?	London / Winnipeg, MB
36	Row35	1	1	Bowen, Mi...	female	45	0	0	PC 17608	262.375	?	C	4	?	Cooperstown, NY
37	Row36	1	1	Bowerman,...	female	22	0	1	113505	55	E33	S	6	?	St Leonards-on-Sea, England Ohio
38	Row37	1	1	Bradley, Mr...	male	?	0	0	111427	26.55	?	S	9	?	Los Angeles, CA
39	Row38	1	0	Brady, Mr. ...	male	41	0	0	113054	30.5	A21	S	?	?	Pomeroy, WA
40	Row39	1	0	Brandeis, ...	male	48	0	0	PC 17591	50.496	B10	C	?	208	Omaha, NE
41	Row40	1	0	Brewe, Dr. ...	male	?	0	0	112379	39.6	?	C	?	?	Philadelphia, PA
42	Row41	1	1	Brown, Mrs...	female	44	0	0	PC 17610	27.721	B4	C	6	?	Denver, CO
43	Row42	1	1	Brown, Mrs...	female	59	2	0	11769	51.479	C101	S	D	?	Belmont, MA
44	Row43	1	1	Bucknell, ...	female	60	0	0	11813	76.292	D15	C	8	?	Philadelphia, PA
45	Row44	1	1	Burns, Mis...	female	41	0	0	16966	134.5	E40	C	3	?	?
46	Row45	1	0	Butt, Major...	male	45	0	0	113050	26.55	B38	S	?	?	Washington, DC
47	Row46	1	0	Cairns, Mr. ...	male	?	0	0	113798	31	?	S	?	?	?
48	Row47	1	1	Calderhea...	male	42	0	0	PC 17476	26.288	E24	S	5	?	New York, NY
49	Row48	1	1	Candee, Mr...	female	53	0	0	PC 17606	27.446	?	C	6	?	Washington, DC
50	Row49	1	1	Cardeza, M...	male	36	0	1	PC 17755	512.329	B51 B53 B55	C	3	?	Austria-Hungary / Germantown, Philadelphia, PA
51	Row50	1	1	Cardeza, M...	female	58	0	1	PC 17755	512.329	B51 B53 B55	C	3	?	Germantown, Philadelphia, PA
52	Row51	1	0	Carlsson, ...	male	33	0	0	695	5	B51 B53 B55	S	?	?	New York, NY
53	Row52	1	0	Carrau, Mr....	male	28	0	0	113059	47.1	?	S	?	?	Montevideo, Uruguay
54	Row53	1	0	Carrau, Mr....	male	17	0	0	113059	47.1	?	S	?	?	Montevideo, Uruguay



IF (% missing value > threshold) THEN remove column

Low Variance

#	Row...	pclass Number (integer)	survived Number (integer)	name String	sex String	age Number (double)	sibsp Number (integer)	parch Number (integer)	ticket String	fare Number (double)	cabin String	embarked String	boat String	body Number (integer)	home.dest String
29	Row28	1	1	Bissette, M...	female	35	0	0	PC 17760	135.633	C99	S	8	?	?
30	Row29	1	1	Bjornstrom...	male	28	0	0	110564	26.55	C52	S	D	?	Stockholm, Sweden / Washington, DC
31	Row30	1	0	Blackwell, ...	male	45	0	0	113784	35.5	T	S	?	?	Trenton, NJ
32	Row31	1	1	Blank, Mr. ...	male	40	0	0	112277	31	A31	C	7	?	Glen Ridge, NJ
33	Row32	1	1	Bonnell, Mi...	female	30	0	0	36928	164.867	C7	S	8	?	Youngstown, OH
34	Row33	1	1	Bonnell, Mi...	female	58	0	0	113783	26.55	C103	S	8	?	Birkdale, England Cleveland, Ohio
35	Row34	1	0	Borebank, ...	male	42	0	0	110489	26.55	D22	S	?	?	London / Winnipeg, MB
36	Row35	1	1	Bowen, Mi...	female	45	0	0	PC 17608	262.375	?	C	4	?	Cooperstown, NY
37	Row36	1	1	Bowerman,...	female	22	0	1	113505	55	E33	S	6	?	St Leonards-on-Sea, England Ohio
38	Row37	1	1	Bradley, Mr...	male	?	0	0	111427	26.55	?	S	9	?	Los Angeles, CA
39	Row38	1	0	Brady, Mr. ...	male	41	0	0	113054	30.5	A21	S	?	?	Pomeroy, WA
40	Row39	1	0	Brandeis, ...	male	48	0	0	PC 17591	50.496	B10	C	?	208	Omaha, NE
41	Row40	1	0	Brewe, Dr. ...	male	?	0	0	112379	39.6	?	C	?	?	Philadelphia, PA
42	Row41	1	1	Brown, Mrs...	female	44	0	0	PC 17610	27.721	B4	C	6	?	Denver, CO
43	Row42	1	1	Brown, Mrs...	female	59	2	0	11769	51.479	C101	S	D	?	Belmont, MA
44	Row43	1	1	Bucknell, ...	female	60	0	0	11813	76.292	D15	C	8	?	Philadelphia, PA
45	Row44	1	1	Burns, Mis...	female	41	0	0	16966	134.5	E40	C	3	?	?
46	Row45	1	0	Butt, Major...	male	45	0	0	113050	26.55	B38	S	?	?	Washington, DC
47	Row46	1	0	Cairns, Mr. ...	male	?	0	0	113798	31	?	S	?	?	?
48	Row47	1	1	Calderhea...	male	42	0	0	PC 17476	26.288	E24	S	5	?	New York, NY
49	Row48	1	1	Candee, Mr...	female	53	0	0	PC 17606	27.446	?	C	6	?	Washington, DC
50	Row49	1	1	Cardeza, M...	male	36	0	1	PC 17755	512.329	B51 B53 B55	C	3	?	Austria-Hungary / Germantown, Philadelphia, PA
51	Row50	1	1	Cardeza, M...	female	58	0	1	PC 17755	512.329	B51 B53 B55	C	3	?	Germantown, Philadelphia, PA
52	Row51	1	0	Carlsson, ...	male	33	0	0	695	5	B51 B53 B55	S	?	?	New York, NY
53	Row52	1	0	Carrau, Mr....	male	28	0	0	113059	47.1	?	S	?	?	Montevideo, Uruguay
54	Row53	1	0	Carrau, Mr....	male	17	0	0	113059	47.1	?	S	?	?	Montevideo, Uruguay

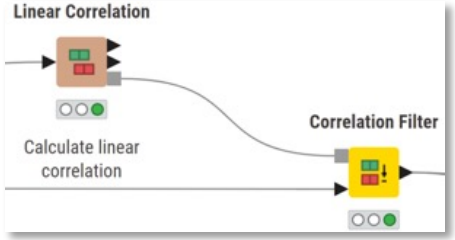


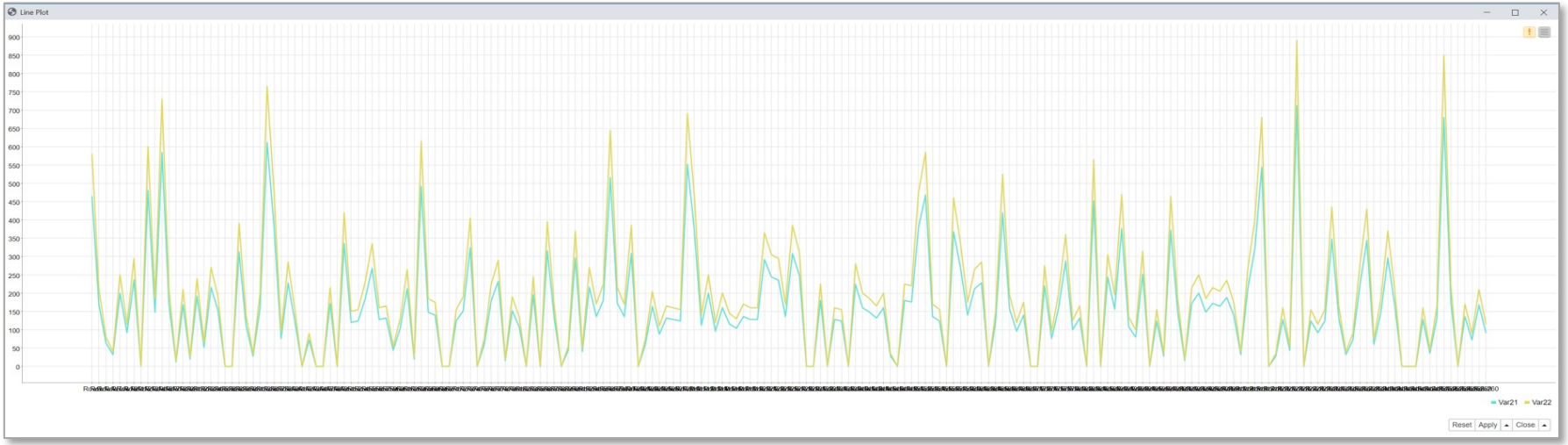
Note: requires min-max-normalization, and only works for numeric columns

- If column has **constant** value (variance = **0**), it contains no useful information
- In general: IF (variance < threshold) THEN remove column

High Correlation

- Two **highly correlated** input variables probably carry similar information
- IF (**corr(var1, var2)** > threshold) => remove var1





Principal Component Analysis (PCA)

- PCA is a statistical procedure that **orthogonally** transforms the original n coordinates of a data set into a new set of n coordinates, called principal components.

$$(PC_1, PC_2, \cdots PC_n) = PCA(X_1, X_2, \cdots X_n)$$

- The first principal component PC_1 follows the direction (eigenvector) of the **largest possible variance** (largest eigenvalue of the covariance matrix) in the data.

- Each succeeding component PC_k follows the direction of the **next largest possible variance** under the constraint that it is orthogonal to (i.e., uncorrelated with) the preceding components $(PC_1, PC_2, \cdots PC_{k-1})$.

If you're still curious, there's LOTS of different ways to think about PCA: https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

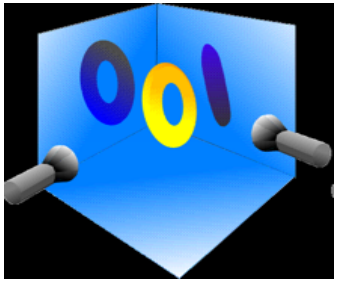
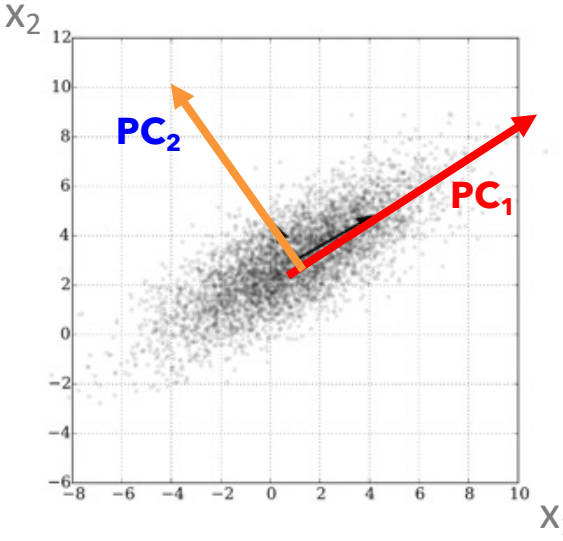


Image from Wikipedia



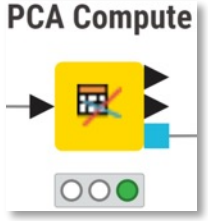
Principal Component Analysis (PCA)

- PC_1 describes most of the variability in the data, PC_2 adds the next big contribution, and so on. In the end, the last PCs do not bring much more information to describe the data.

- Thus, to describe the data we could use only the top $m < n$ (i.e., $PC_1, PC_2, \cdots PC_m$) components with little - if any - loss of information

- Caveats:
  - Results of PCA are quite difficult to interpret
  - Normalization required
  - Only effective on numeric columns

Dimensionality Reduction

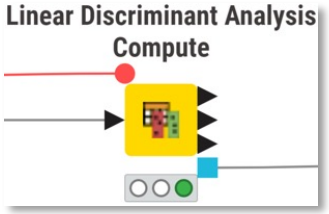


Linear Discriminant Analysis (LDA)

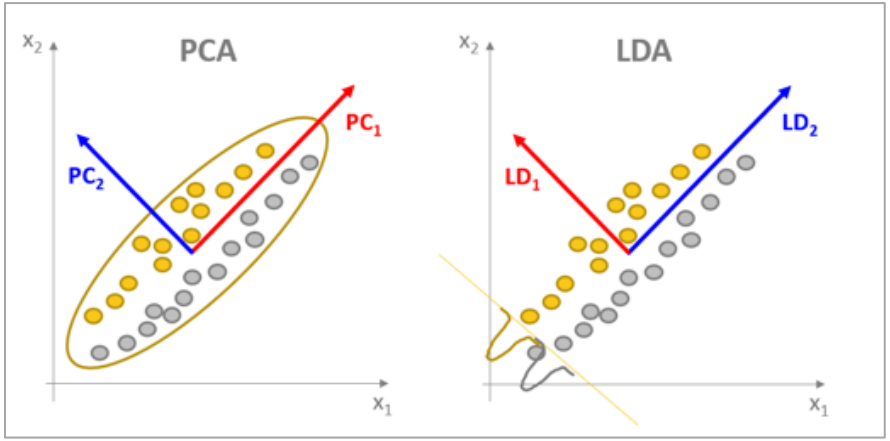
- LDA is a statistical procedure that **orthogonally** transforms the original n coordinates of a data set into a new set of k-1 coordinates, called linear discriminants, where k is the number of classes in the class variable

$$(LD_1, LD_2, \cdots LD_{k-1}) = LDA(X_1, X_2, \cdots X_n)$$

- Here, however, discriminants (components) **maximize the separation between classes**



- PCA : unsupervised
- LDA : supervised



Linear Discriminant Analysis (LDA)

- LD_1 describes best the class separation in the data, LD_2 adds the next big contribution, and so on. In the end, the last LDs do not bring much more information to separate the classes.

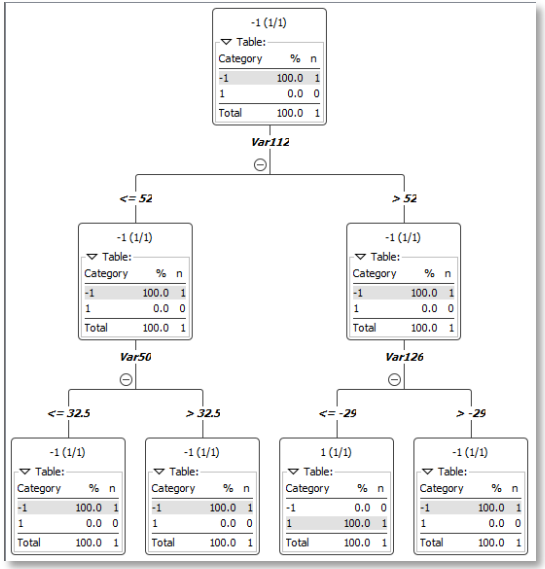
- Thus, for our classification problem we could use only the top $m < k - 1$ (i.e., $LD_1, LD_2, \cdots LD_m$) discriminants with little - if any - loss of information

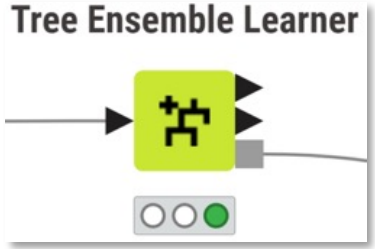
- Caveats:
  - Results of LDA are quite difficult to interpret
  - Normalization required
  - Only effective on numeric columns

Dimensionality Reduction

Ensembles of Shallow Decision Trees

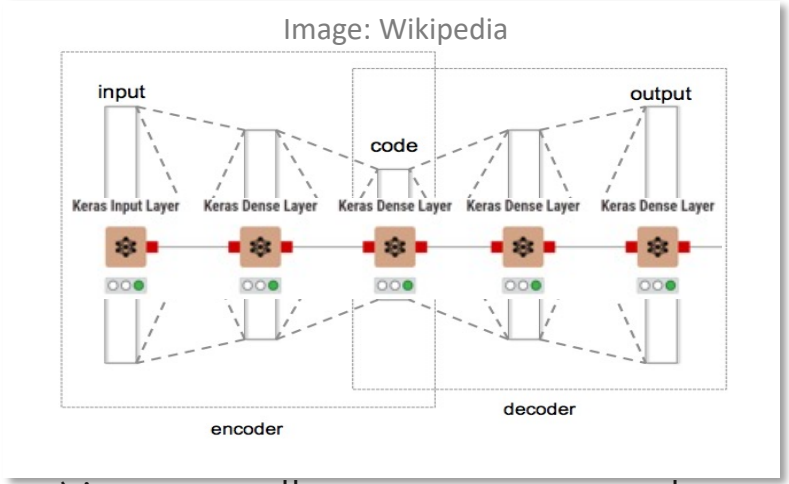
- Often used for classification, but can be used for feature selection too
- Generate a large number (we used 2000) of trees that are very shallow (2 levels, 3 sampled features)
- Calculate the statistics of candidates and selected features. The more often a feature is selected in such trees, the more likely it contains predictive information
- Compare the same statistics with a forest of trees trained on a random dataset.





Autoencoder

- Feed-Forward Neural Network architecture with encoder / decoder structure.
  The network is trained to reproduce the input vector onto the output layer.



- That is, it compresses the input vector (dimension n) into a smaller vector space on layer "code" (dimension m<n) and then it reconstructs the original vector onto the output layer.

- If the network was trained well, the reconstruction operation happens with minimal loss of information.

Data Preparation: Feature Selection

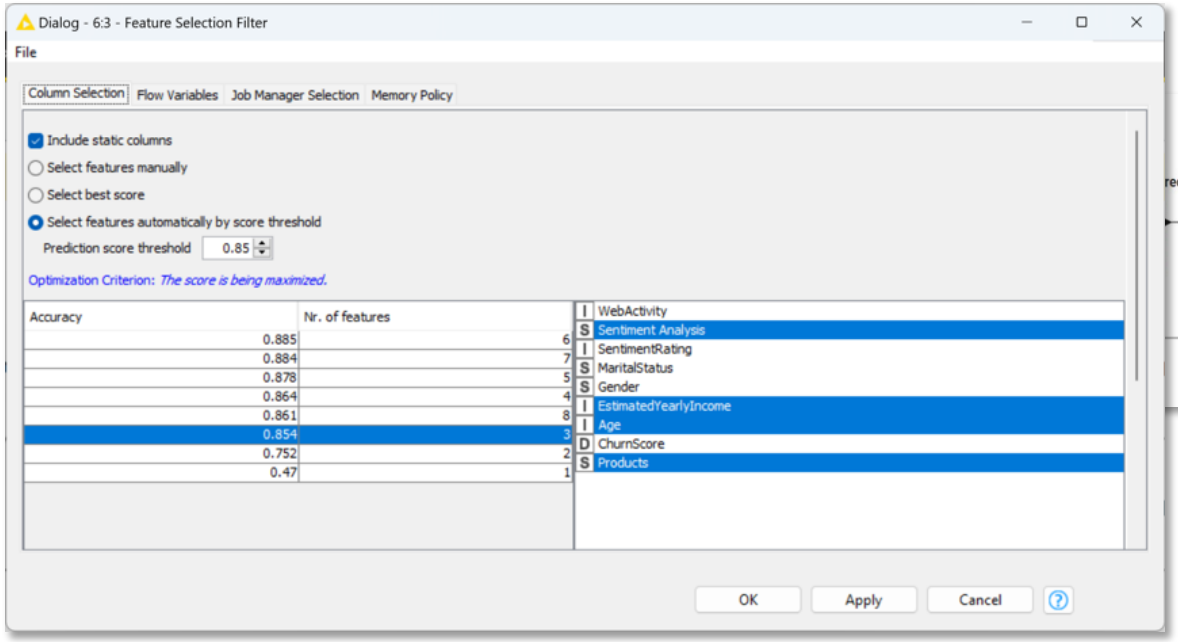
Feature Selection vs. Dimensionality Reduction

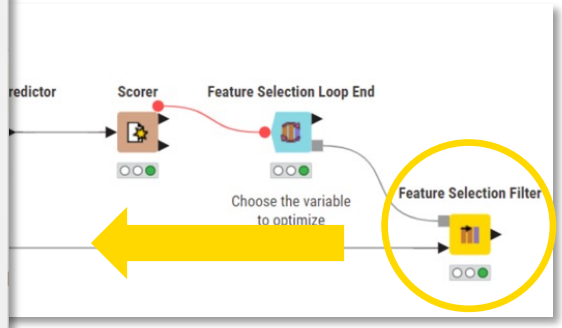
- Both methods are used for reducing the number of features in a dataset. However:
- Feature selection is simply selecting and excluding given features **without changing** them.
- Dimensionality reduction **might transform** the features into a lower dimension.
- Feature selection is often a somewhat more aggressive and more computationally expensive process.
  - Backward Feature Elimination
  - Forward Feature Construction

Backward Feature Elimination (greedy top-down)

1. First train one model on n input features
2. Then train n separate models each on $n-1$ input features and remove the feature whose removal produced the least disturbance
3. Then train $n-1$ separate models each on $n-2$ input features and remove the feature whose removal produced the least disturbance
4. And so on. Continue until desired maximum error rate on *training* data is reached.

Backward Feature Elimination

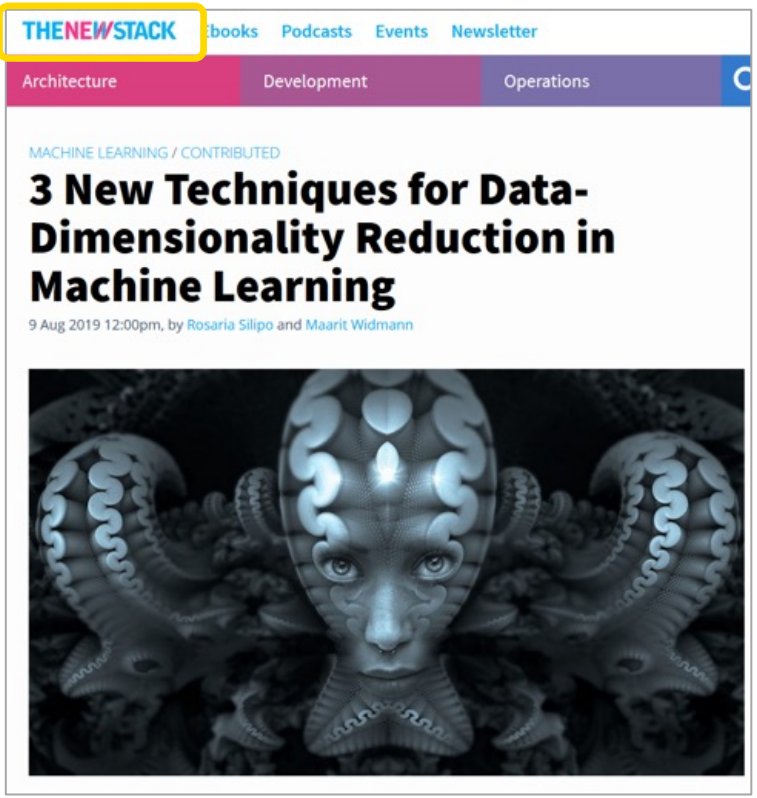




Forward Feature Construction (greedy bottom-up)

1. First, train n separate models on one single input feature and keep the feature that produces the best accuracy.
2. Then, train $n - 1$ separate models on 2 input features, the selected one and one more. At the end keep the additional feature that produces the best accuracy.
3. And so on … Continue until an acceptable error rate is reached.

Material

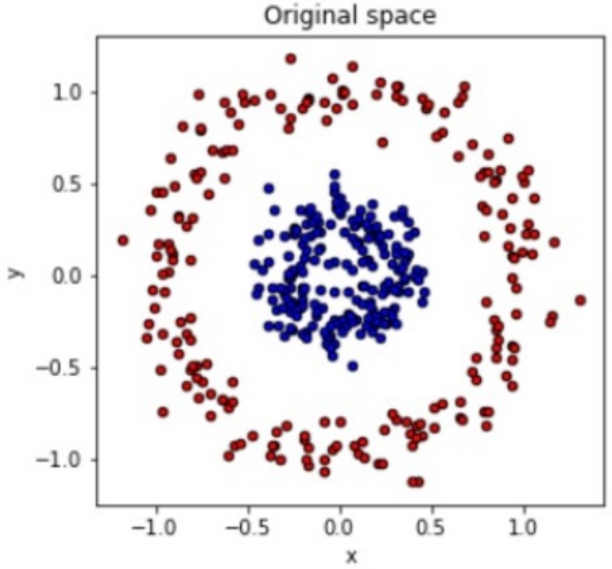


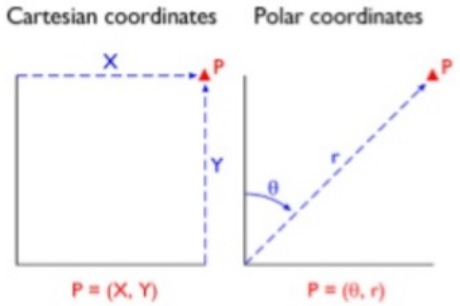
https://thenewstack.io/3-new-techniques-for-data-dimensionality-reduction-in-machine-learning/

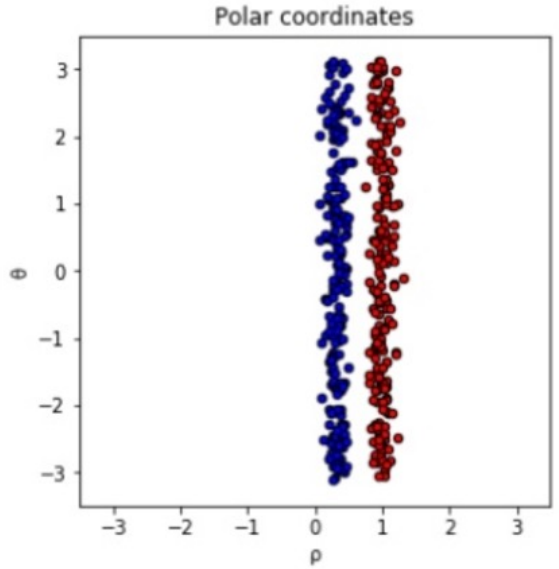
Data Preparation: Feature Engineering

Feature Engineering: Motivation

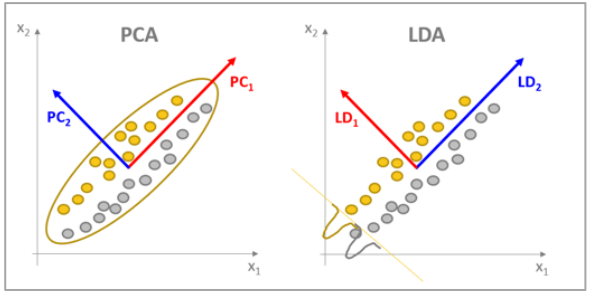
Sometimes transforming the original data allows for better discrimination by ML algorithms.







Feature Engineering: Techniques



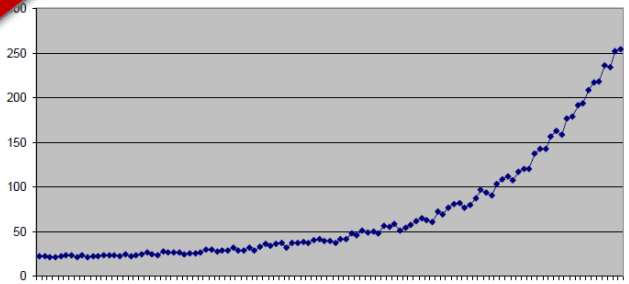
- Coordinate Transformations
  Remember PCA and LDA?
  Polar coordinates , …

- Distances to cluster centres, after data clustering
- Simple math transformations on single columns
  $(e^x, x^2, x^3, \tanh(x), \log(x)\ , \ldots)$
- Combining together multiple columns in math functions
  $(f(x_1, x_2, \ldots\ xn), x_2 - x_1, \ldots)$
- The whole process is domain dependent

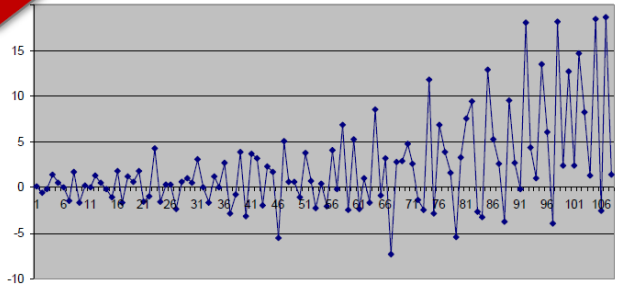
Feature Engineering in Time Series Analysis

- Second order differences: $y = x(t) - x(t-1)$ & $y'(t) = y(t) - y(t-1)$
- Logarithm: $\log(y'(t))$

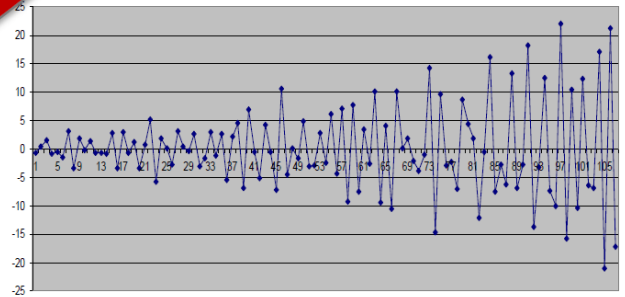
1 - Original Time Series: non-stationary (mean and variance)



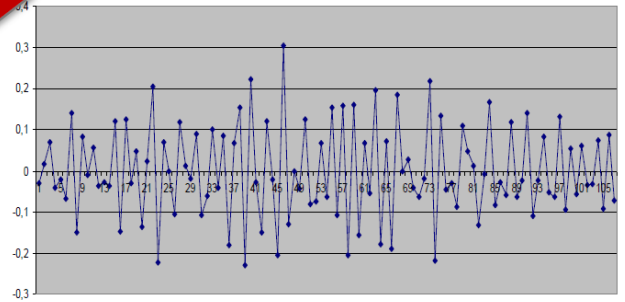
2 – First Order Differencing: non-stationary (mean and variance)



3 – Second Order Diff.: stationary in mean, but not in variance



4 – Double Differencing applied to Log(Series): stationary series



Regression Problems

Supervised Learning: Classification vs. Regression

- $\boldsymbol{X} = (x_1, x_2)$ and $y = \{label\ 1, \ldots, label\ n\}$ or $y \in \mathbb{R}$
- A training set with many examples of $(\boldsymbol{X}, y)$
- The model learns on the examples of the training set to produce the right value of y for an input vector $\boldsymbol{X}$

Classification

y = {yellow, gray}
y = {churn, no churn}
y = {increase, unchanged, decrease}
y = {blonde, gray, brown, red, black}
y = {job 1, job 2, … , job n}

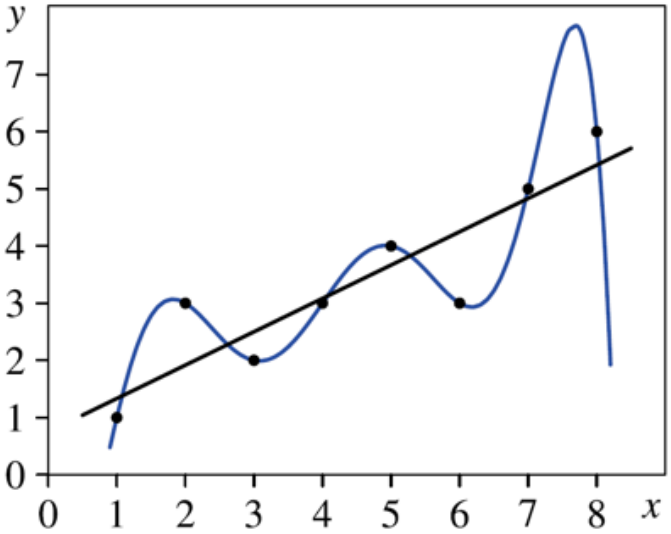
Numerical Predictions (Regression)

y = temperature
y = number of visitors
y = number of kW
y = price
y = number of hours

Regression Overview

- Goal: Explain how target attribute depends on descripitive attributes
  - Target attribute ➔ ***Response variable***
  - Descriptive attribute(s) ➔ ***Regressor variable(s)***

- As a parameterized function class *f*
  - Estimate parameters to describe the relationship
  - Must be simple enough for interpolation and extrapolation purposes
  - Example:
    Line (black) v.s. Polynomial (blue) with degree 7



Regression

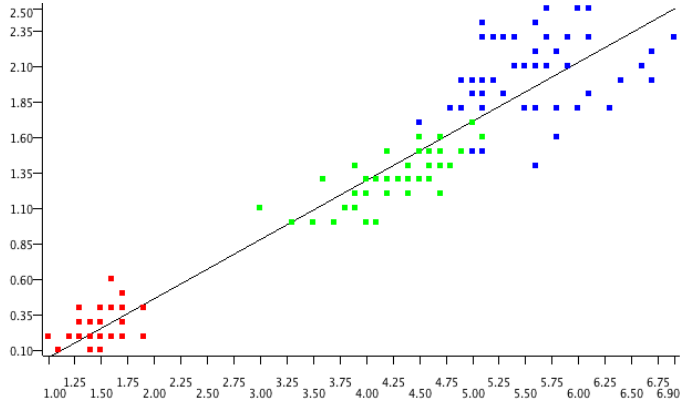
Predict *numeric* outcomes on existing data (supervised)

Applications

- Forecasting
- Quantitative Analysis

Methods

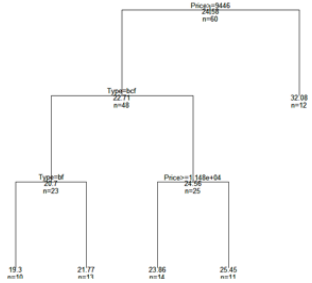
- Linear
- Polynomial
- Regression Trees
- Partial Least Squares



Statistics on Linear Regression

Variable	Coeff.	Std. Err.	t-value	P>|t|
Petal.Length	0.4158	0.0096	43.3872	0.0
Intercept	-0.3631	0.0398	-9.1312	4.44E-16

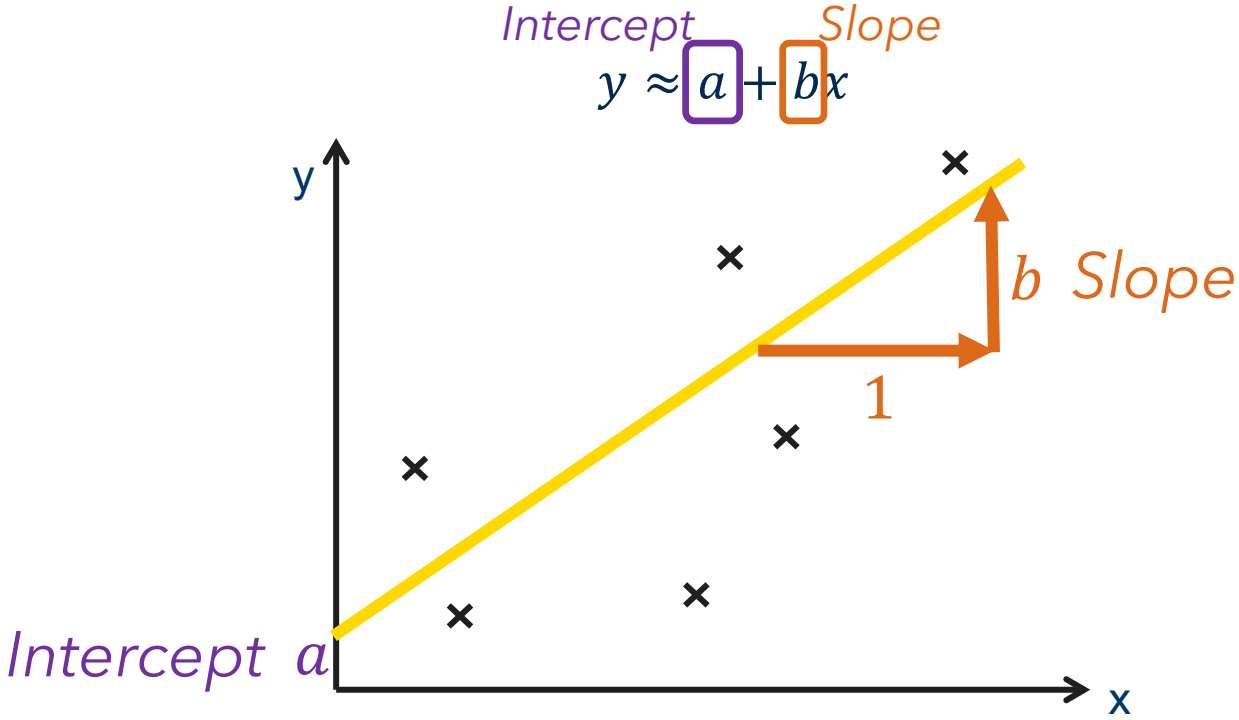
Multiple R-Squared: 0.9271
Adjusted R-Squared: 0.9266



Linear Regression Algorithm

Regression Line

- Given a data set with two continuous attributes, x and y
- There is an approximate linear dependency between x and y



Regression Line

- Given a data set with two continuous attributes, x and y
- There is an approximate linear dependency between x and y

Intercept *Slope*

$$y \approx a + bx$$

- We find a **regression line** (i.e., determine the parameters a and b) such that the line fits the data as well as possible

- Examples:
  - Trend estimation (e.g., oil price over time)
  - Epidemiology (e.g., cigarette smoking vs. lifespan)
  - Finance (e.g., return on investment vs. return on all risky assets)
  - Economics (e.g., spending vs. available income)

Linear Regression

Predicts the values of the target variable y
based on a linear combination of
the values of the input feature(s) x_j

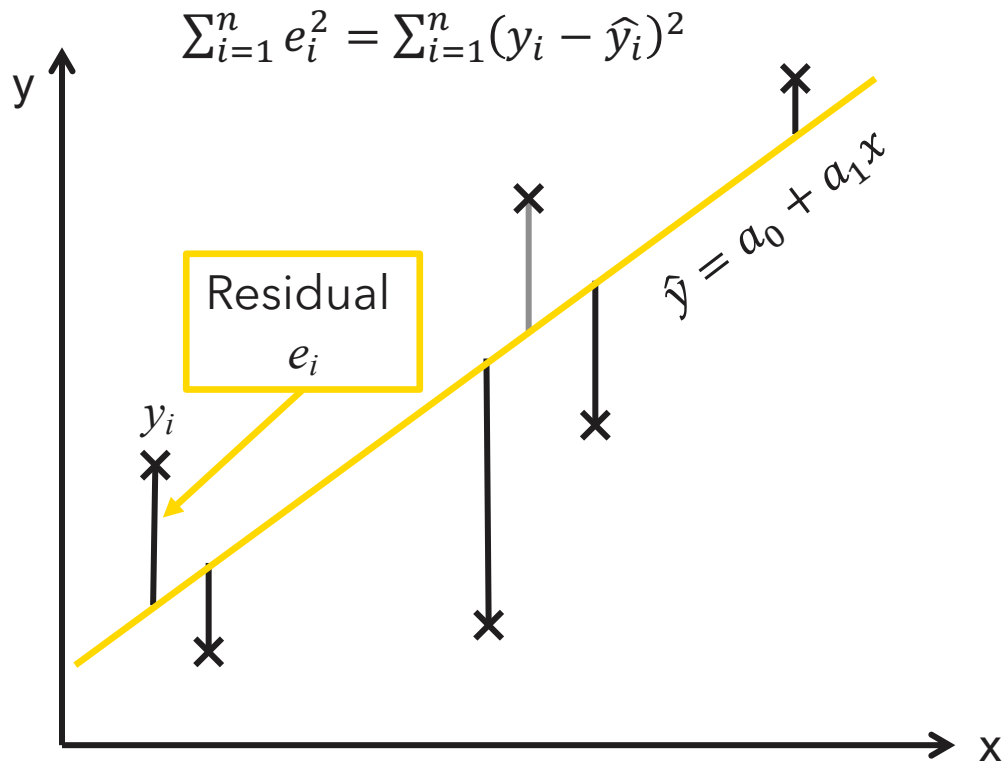
Two input features: $\hat{y} = a_0 + a_1 x_1 + a_2 x_2$

p input features: $\hat{y} = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_p x_p$

- Simple regression: one input feature → regression line
- Multiple regression: several input features → regression hyper-plane
- Residuals: differences between observed and predicted values (errors)
  Use the residuals to measure the model fit

Simple Linear Regression

Optimization goal: minimize sum of squared residuals



Simple Linear Regression

- Think of a straight line $\hat{y} = f(x) = a + bx$
- Find a and b to model all observations (x_i, y_i) as close as possible

  ➔ SSE $F(a,b) = \sum_{i=1}^{n}(f(x) - y_i)^2 = \sum_{i=1}^{n}(a + bx_i - y_i)^2$ should be minimal
- That is:

$$\frac{\partial F}{\partial a} = \sum_{i=1}^{n} 2(a + bx_i - y_i) = 0$$

$$\frac{\partial F}{\partial b} = \sum_{i=1}^{n} 2(a + bx_i - y_i)\, x_i = 0$$

➔ A unique solution exists for a and b

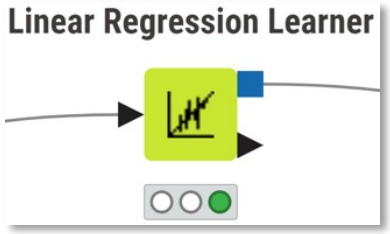
Linear Regression

- Optimization goal: minimize the squared residuals

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{n} a_j x_{j,i} \right)^2 = (y - aX)^T (y - aX)$$

- Solution:

$$\hat{a} = (X^T X)^{-1} X^T y$$



- Computational issues:
  - $X^T X$ must have full rank, and thus be invertible
    (Problems arise if linear dependencies between input features exist)
  - Solution may be unstable, if input features are almost linearly dependent

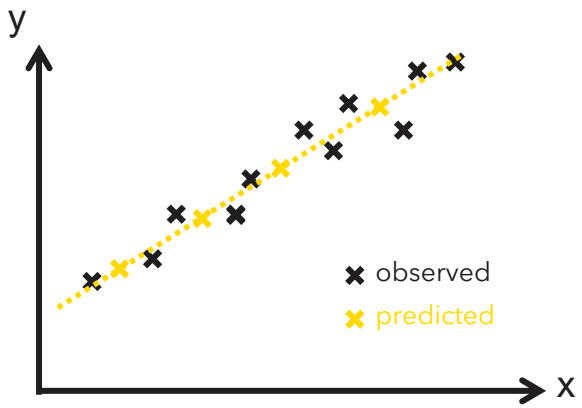
Linear Regression: Summary

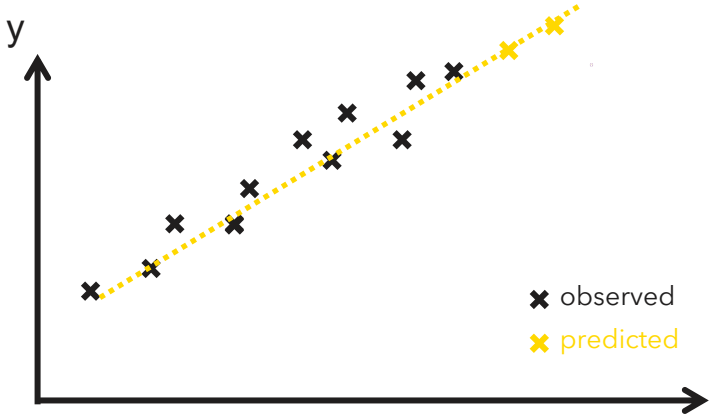
- Positive:
  - Strong mathematical foundation
  - Simple to calculate and to understand
    (For moderate number of dimensions)
  - High predictive accuracy
    (In many applications)

- Negative:
  - Many dependencies are non-linear
    (Can be generalized)
  - Model is global and cannot adapt well to locally different data distributions
    But: Locally weighted regression, CART

Regression vs. Time Series Analysis

- Regression
  - Targets & set of input features
  - Describing the relationship between the target and input features
  - Model → interpolation
- Time series analysis
  - Sequence of observations
  - Predicting future obs from
    - Existing time series
    - Accompanying time series
  - Model → extrapolation





Polynomial Regression

Predicts the values of the target variable y
based on a polynomial combination of degree d of
the values of the input feature(s) x_j

$$\tilde{y} = a_0 + \sum_{j=1}^{p} a_{j,1} x_j + \sum_{j=1}^{p} a_{j,2} x_j^2 + \cdots + \sum_{j=1}^{p} a_{j,d} x_j^d$$

- Simple regression: one input feature → regression curve
- Multiple regression: several input features → regression hypersurface
- Residuals: differences between observed and predicted values (errors)
  Use the residuals to measure the model fit

Evaluation of Regression Models

Numeric Errors: Formulas

Error Metric	Formula	Notes
R-squared	$1 - \frac{\sum_{i=1}^{n}(y_i - f(x_i))^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$	Universal range: the closer to 1 the better
Mean absolute error (MAE)	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - f(x_i)\rvert$	Equal weights to all distances Same unit as the target column
Mean squared error (MSE)	$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$	Common loss function
Root mean squared error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2}$	Weights big differences more Same unit as the target column
Mean signed difference	$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))$	Only informative about the direction of the error
Mean absolute percentage error (MAPE)	$\frac{1}{n}\sum_{i=1}^{n}\frac{\lvert y_i - f(x_i)\rvert}{\lvert y_i\rvert}$	Requires non-zero target column values

MAE (Mean Absolute Error) vs. RMSE (Root Mean Squared Error)

MAE	RMSE
Easy to interpret – mean absolute error	Cannot be directly interpreted as the average error
All errors are equally weighted	Larger errors are weighted more
Generally smaller than RMSE	Ideal when large deviations need to be avoided

Example:

Actual values = [2,4,5,8],

Case 1: Predicted Values = [4, 6, 8, 10]

Case 2: Predicted Values = [4, 6, 8, 14]

	MAE	RMSE
Case 1	2.25	2.29
Case 2	3.25	3.64

R-squared vs. RMSE

R-squared	RMSE
Relative measure: Proportion of variability explained by the model	**Absolute measure**: How much deviation at each point
Range: Usually between 0 and 1. 0 = no variability explained 1 = all variability explained	Same scale as the target

Example:

Actual values = [2,4,5,8],

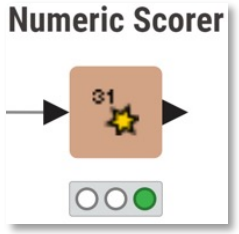
Case 1: Predicted Values = [3, 4, 5, 6]

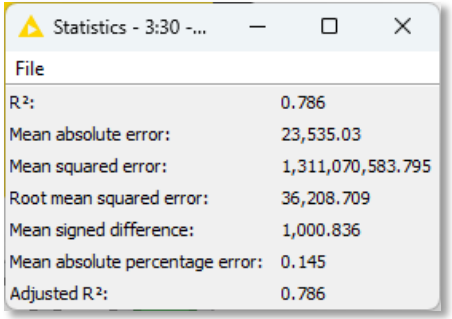
Case 2: Predicted Values = [3, 3, 7, 7]

	R-sq	RMSE
Case 1	0.96	1.12
Case 2	0.65	1.32

Numeric Scorer

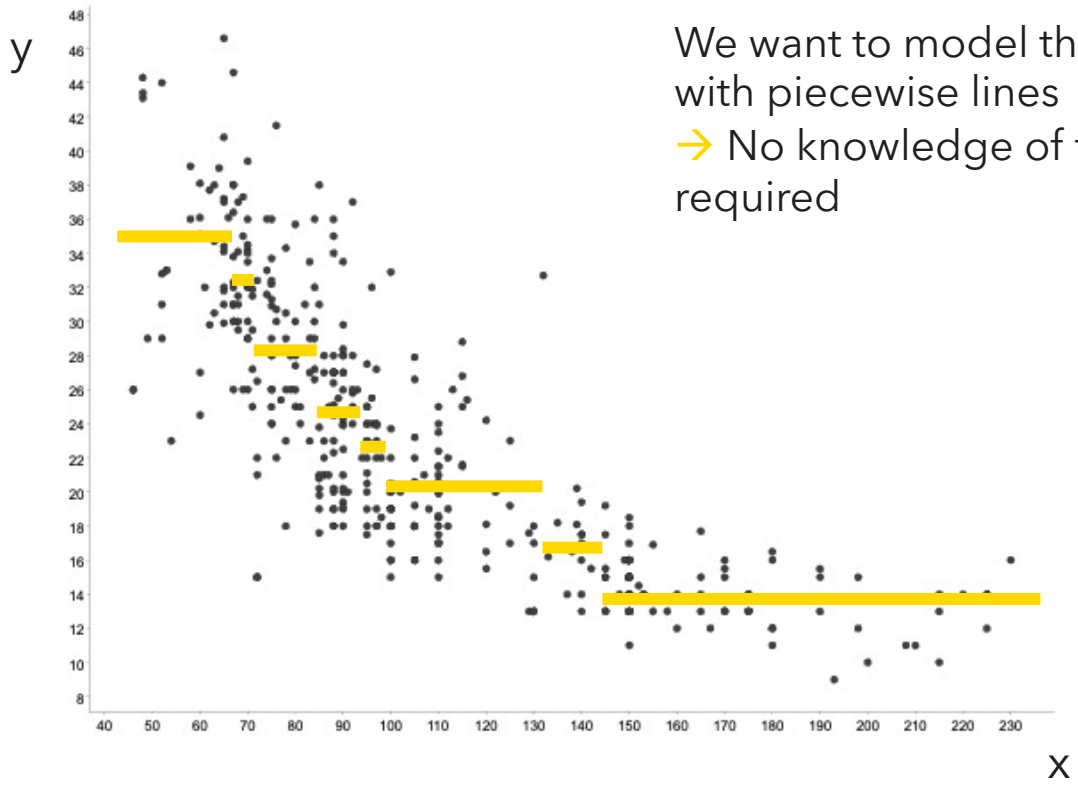
- Similar to scorer node, but for nodes with *numeric* predictions
- Compare dependent variable values to predicted values to evaluate model quality.
- Report R^2, RMSE, MAPE, etc.





Regression Tree

Regression Tree: Goal



We want to model the target variable with piecewise lines

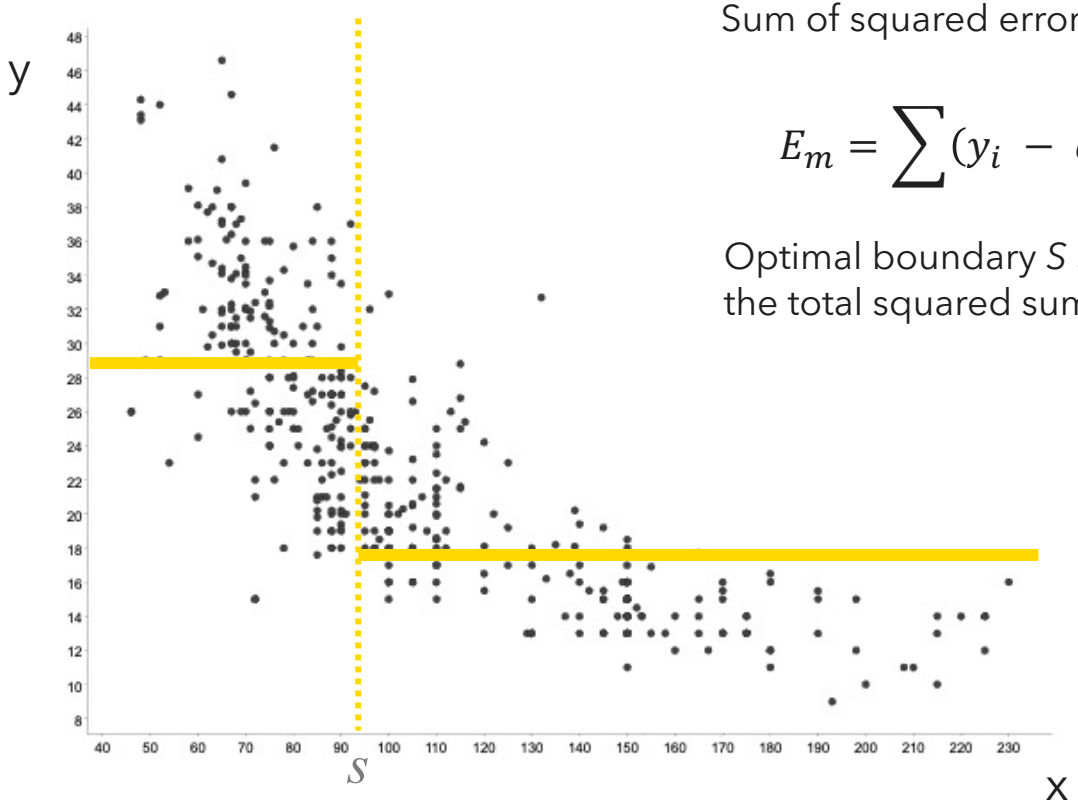
→ No knowledge of functional form required

Regression Tree: Initial Split

Local mean:

$$c_m = \frac{1}{n}\sum y_i$$

For observations in segment *m*



Sum of squared errors:

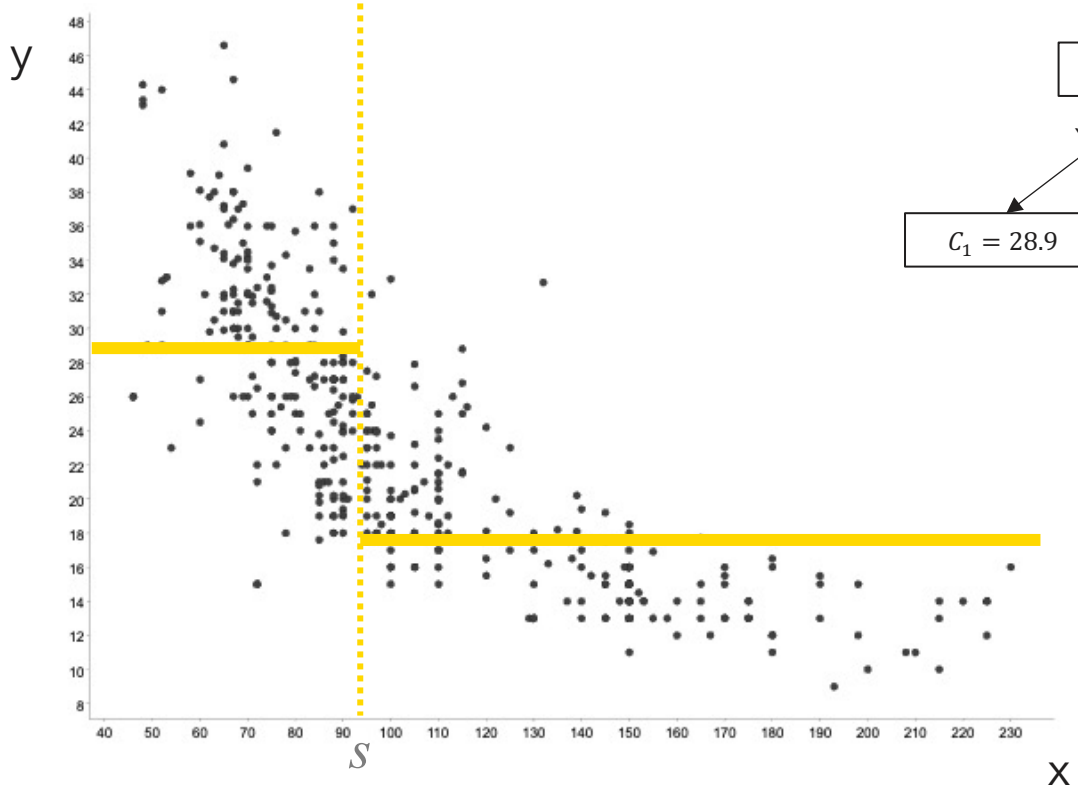
$$E_m = \sum (y_i - c_m)^2$$

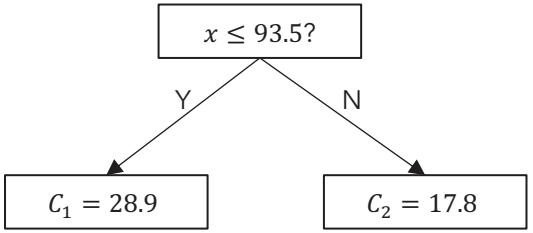
Optimal boundary *S* should minimize the total squared sum:

$$\sum E_m$$

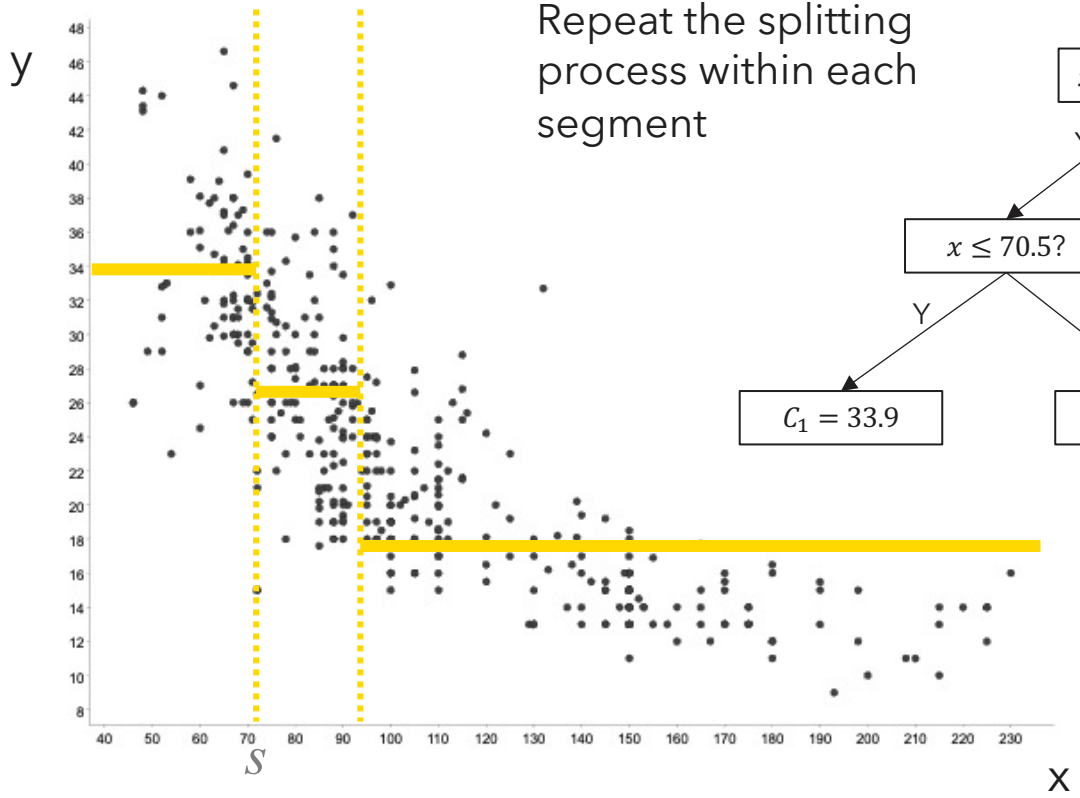
For all segments *m*

Regression Tree: Initial Split

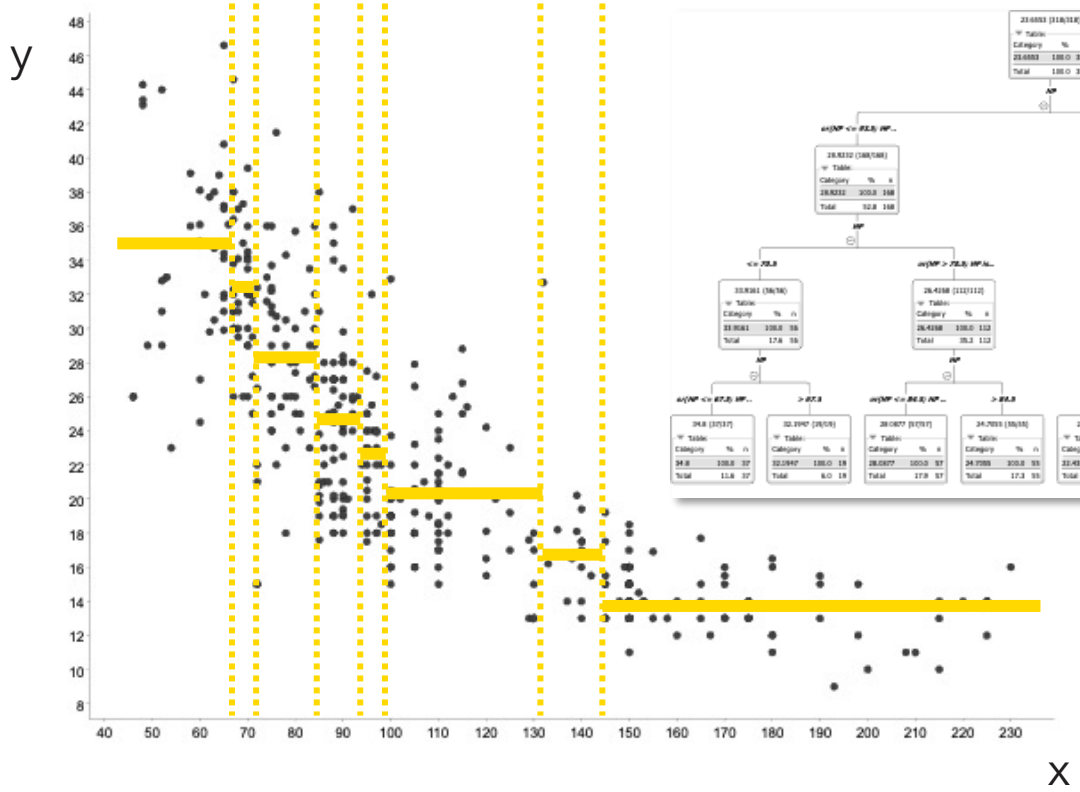




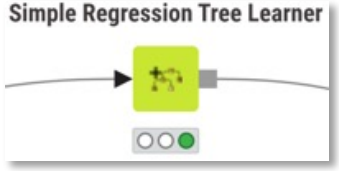
Regression Tree: Growing the Tree



Regression Tree: Final Model



Regression Tree: Algorithm



Start with a single node containing all points.

1. Calculate c_i and E_i.
2. If all points have the same value for feature x_j, stop.
3. Otherwise, find the best binary splits that reduces $E_{j,s}$ as much as possible.
   - $E_{j,s}$ doesn't reduce as much → stop
   - A node contains less than the minimum node size → stop
   - Otherwise, take that split, creating two new nodes.
   - In each new node, go back to step 1.

Regression Trees: Summary

- Differences to decision trees:
  - Splitting criterion: minimizing intra-subset variation (error)
  - Pruning criterion: based on numeric error measure
  - Leaf node predicts average target values of training instances reaching that node

- Can approximate piecewise constant functions
- Easy to interpret

Regression Trees: Pros & Cons

- Finding of (local) regression values (average)
- Problems:
  - No interpolation across borders
  - Heuristic algorithm: unstable and not optimal.

- Extensions:
  - Fuzzy trees (better interpolation)
  - Local models for each leaf (linear, quadratic)

Exercises

Exercises

- Dataset: Sales data of individual residential properties in Ames, Iowa from 2006 to 2010
- One of the columns is the price for which the house was sold
- Goal: Predicting the house **price**

- Data Preparation:
  - 01_Missing_Value_Handling
  - 02_Outlier_Detection
- Regression:
  - 03_Linear_Regression
  - 04_Regression_Trees