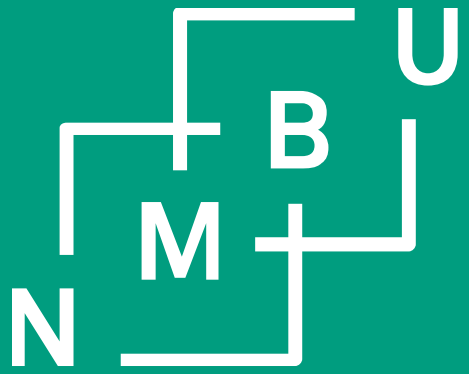
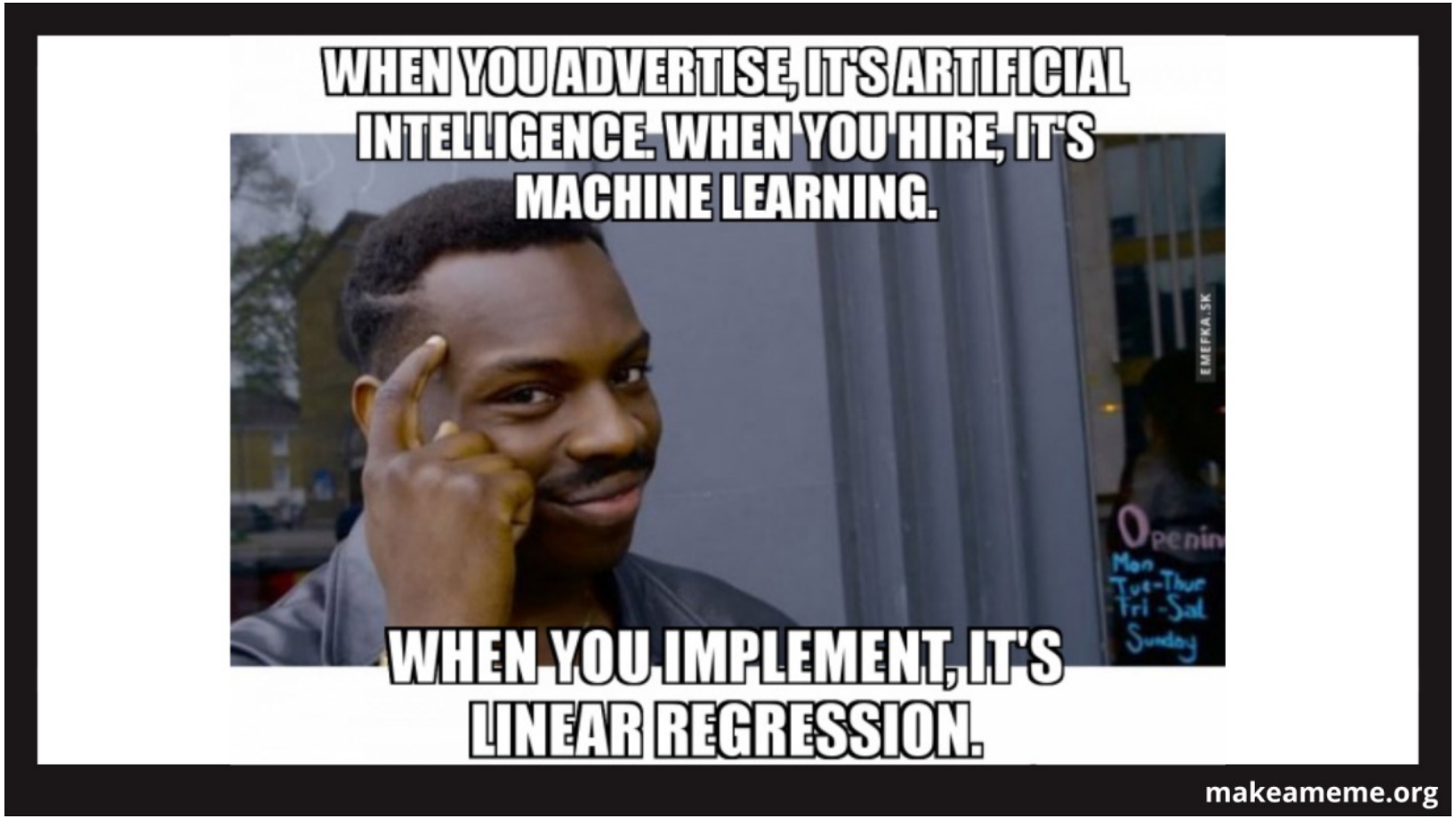


Norwegian University
of Life Sciences

Regression – predicting continuous target variables

Chapter 10 in Python Machine Learning TE


WHEN YOU ADVERTISE, IT'S ARTIFICIAL INTELLIGENCE. WHEN YOU HIRE, IT'S MACHINE LEARNING.
WHEN YOU IMPLEMENT, IT'S LINEAR REGRESSION.
makeameme.org

Summary

- Regression
- Visualisation, pre-checking data quality

- RANSAC
- Performance
- High dimensions and PCR/PLS

- Regularization
- Polynomials and transformations

- Tree based regression
- Regression as the last step (compression, feature selection, …)

Resources

- Python Machine Learning TE, Chapter 10, pages 309 – 345
  - Jupyter notebook : Chapter 10, part 1 + 2

- Ordinary least squares in scikit-learn: http://scikit-learn.org/stable/modules/linear_model.html#ordinary-least-squares
- Robustness and RANSAC in scikit-learn: http://scikit-learn.org/stable/modules/linear_model.html#robustness-regression-outliers-and-modeling-errors
- Decision tree regression in scikit-learn: http://scikit-learn.org/stable/modules/tree.html
- Random forests in scikit-learn: http://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees

Linear regression

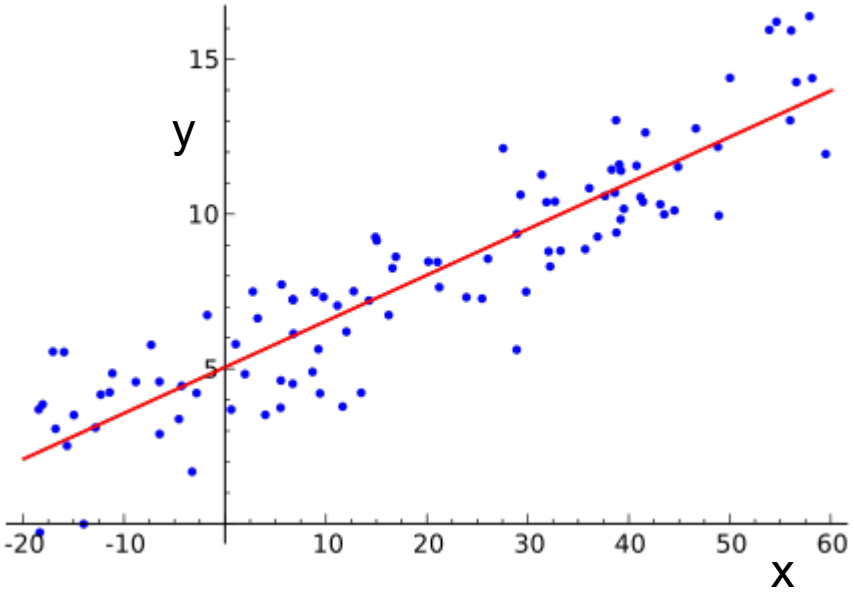
- Modelling and prediction of a continuous target variable

- Supervised learning, i.e., we exploit our knowledge about the target/response

- In statistics, linear regression models share:

$$\boldsymbol{X}w = \boldsymbol{y} + \boldsymbol{\varepsilon}$$

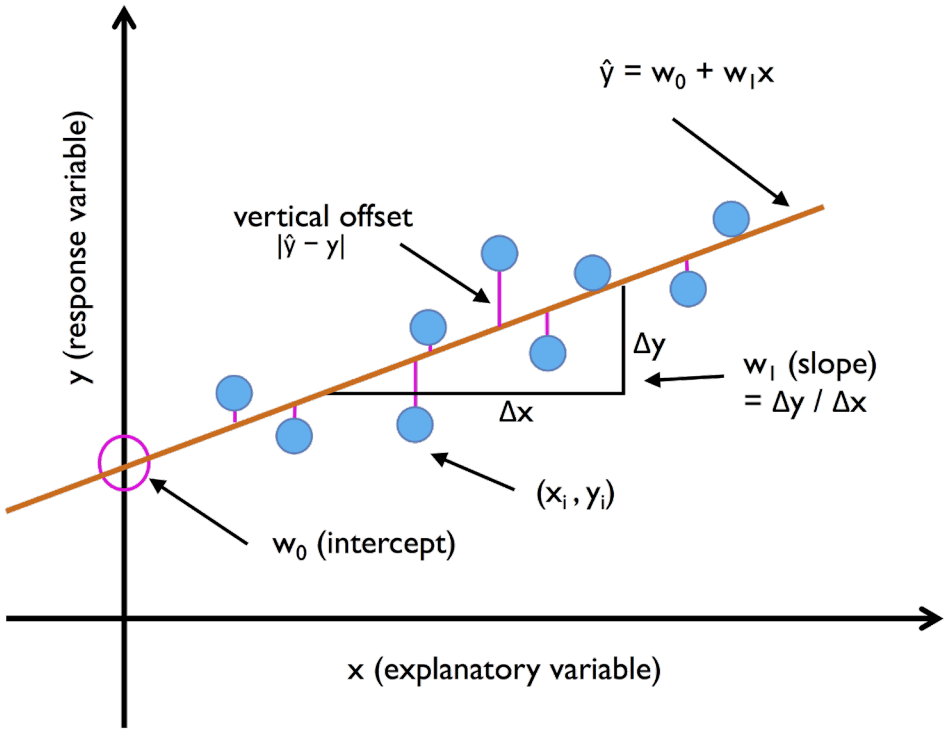
  - $\boldsymbol{X}$: predictor matrix
  - w: weight vector (regression coefficients)
  - $\boldsymbol{y}$: target/response vector
  - $\boldsymbol{\varepsilon}$: error (residual when fitted)



By Sewaqu - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=11967659

Linear regression

- Intercept and one predictor ($[1, x]$), one target (y):
  - a line in a plane
  - $y = w_0 + w_1 x$

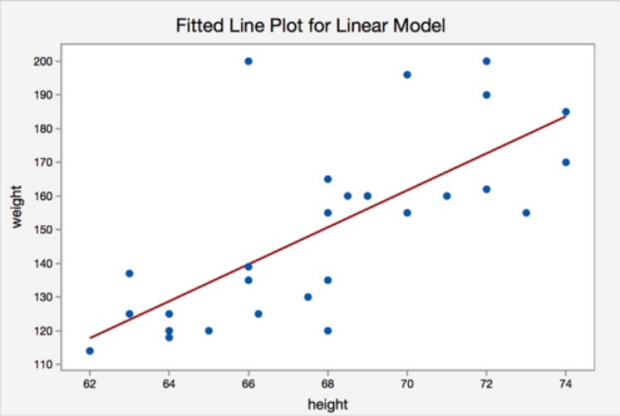


Jupyter: simple_linear_regression.ipynb

(Statistical) Assumptions for Linear regression

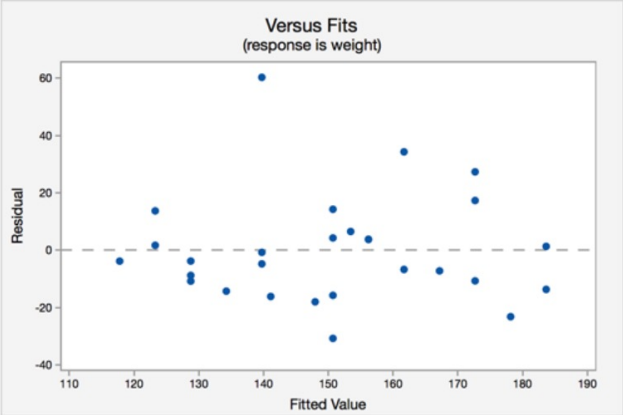
- Linearity
  - The relationship between the variables must be linear.
- Independence of errors
  - There is not a relationship between the residuals and the variable; in other words, is independent of errors.
- Normality of errors
  - The residuals must be approximately normally distributed
- Equal variances
  - The variance of the residuals is the same for all values of the response values..

Assumption 1: Linearity - The relationship between height and weight must be linear.



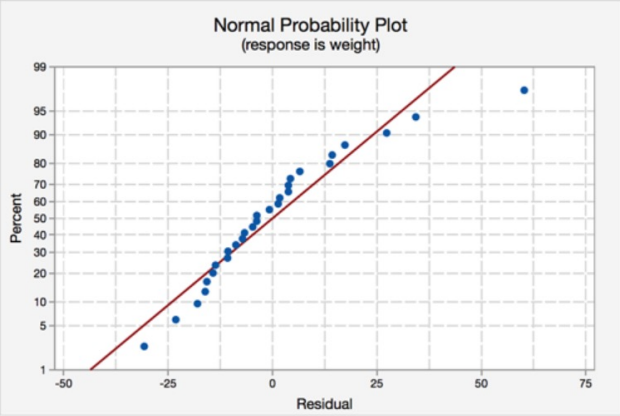
The scatterplot shows that, in general, as height increases, weight increases. There does not appear to be any clear violation that the relationship is not linear.

Assumption 2: Independence of errors - There is not a relationship between the residuals and weight.



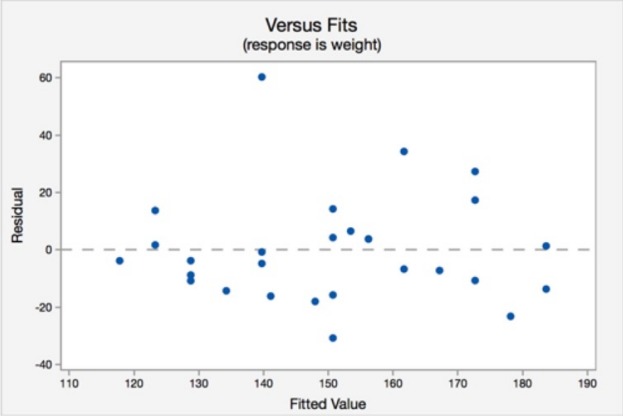
In the residuals versus fits plot, the points seem randomly scattered, and it does not appear that there is a relationship.

Assumption 3: Normality of errors - The residuals must be approximately normally distributed.



Most of the data points fall close to the line, but there does appear to be a slight curving. There is one data point that stands out.

Assumption 4: Equal Variances - The variance of the residuals is the same for all values of X.

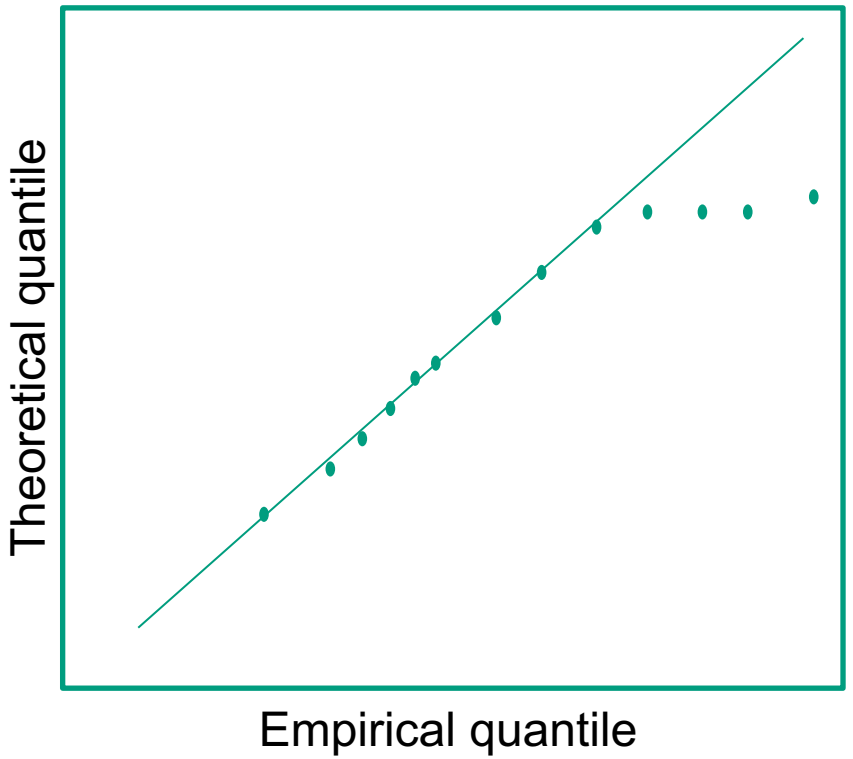


In this plot, there does not seem to be a pattern.

https://online.stat.psu.edu/stat500/lesson/9/9.5

Model assumptions

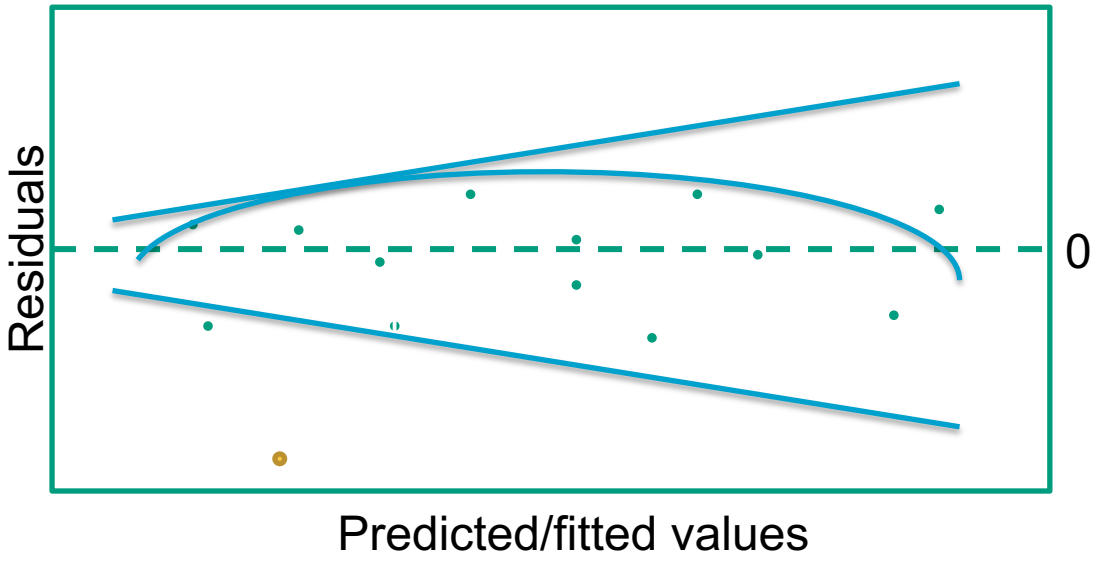
- Typical assumptions in statistics:
  - Normally distributed errors
- Large deviations indicates something is not well described



Model assumptions & diagnostic plotting

- Typical assumptions often emphasized in statistical analysis:
  - Homoscedastic noise*
  - Un-correlated errors

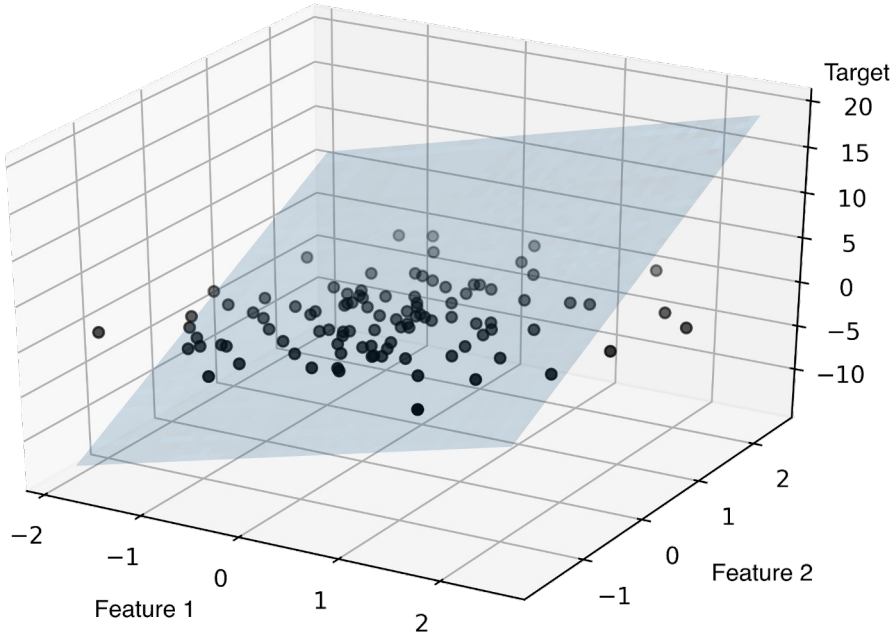
*heteroscedasticity occurs when the size of the error term differs across the independent variable's value



- Strong/systematic patterns can reveal potential for improvements in transforming/pre-processing the data.
- Outliers can be seen by their large residuals (vertical deviations from the "0-line").
  - In statistics residuals are often standardised, and thresholds based on 2-3 standard deviations are used to detect outliers.

Multiple linear regression

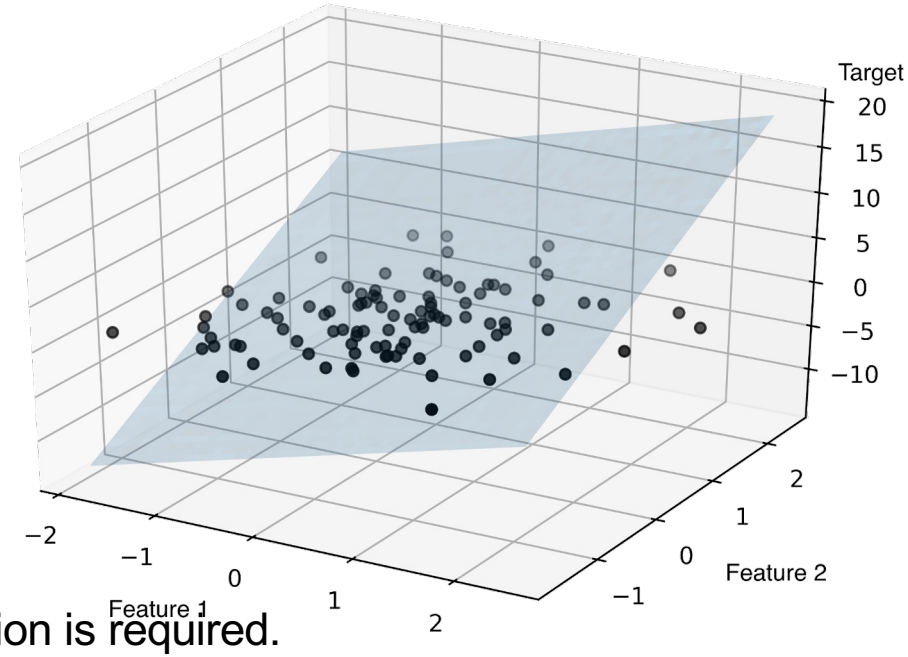
- $p \geq 2$, i.e., more predictors, one target
- Solution: a hyperplane in $p+1$ dimensions
- $\hat{y} = \hat{w}_0 + \hat{w}_1 x + \cdots + \hat{w}_p x = [1, \boldsymbol{x}]\hat{\boldsymbol{w}}$



- As long as the data matrix $\boldsymbol{X}'\boldsymbol{X}$ is invertible (non-singular), the least squares solution $\hat{\boldsymbol{w}}$ is unique.
  - Otherwise some kind of regularization, feature transformation or feature selection is required.

Solving linear regression

- $\hat{y} = \hat{w}_0 + \hat{w}_1 x + \cdots + \hat{w}_p x = [1, \boldsymbol{x}]\hat{\boldsymbol{w}}$
- $\hat{y} = \hat{w}_0 + \hat{w}_1 x = \boldsymbol{x}^T \hat{\boldsymbol{w}}$

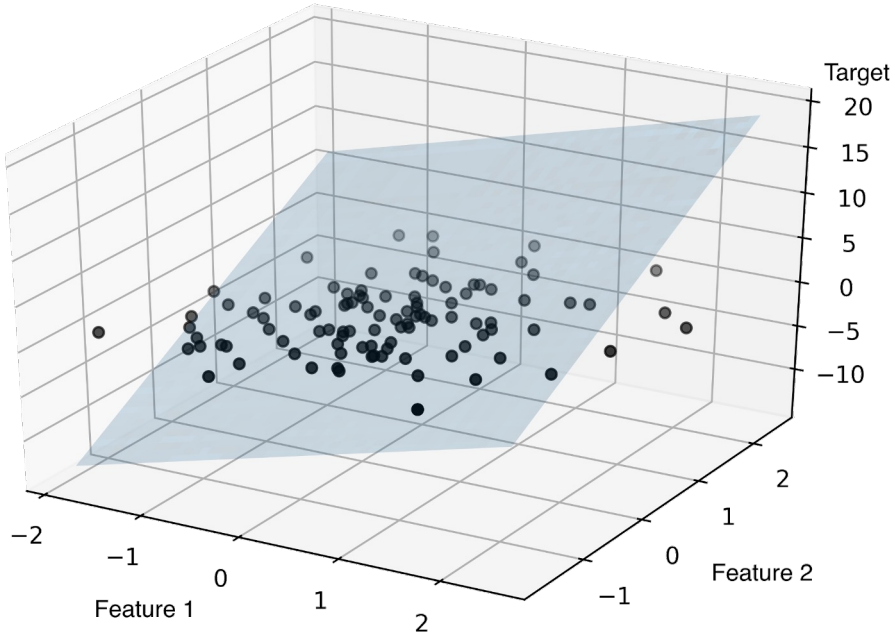


- As long as the data matrix $\boldsymbol{X}'\boldsymbol{X}$ is invertible (non-singular), the least squares solution $\hat{\boldsymbol{w}}$ is unique.
  - Otherwise some kind of regularization, feature transformation or feature selection is required.

Multiple linear regression

- Intercept and several (p) predictors ($[1, \boldsymbol{x}]$), one target (y):

$$\text{Data: } \boldsymbol{X} = \begin{bmatrix} 1 & \boldsymbol{x}_1 \\ \vdots & \vdots \\ 1 & \boldsymbol{x}_n \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$



- Find the least squares solution of:

$$\boldsymbol{Xw} = \boldsymbol{y} + \boldsymbol{\varepsilon}$$

$$||\boldsymbol{X}\hat{\boldsymbol{w}} - \boldsymbol{y}|| \quad \text{(i.e. minimize)}$$

$$\boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{w}} = \boldsymbol{X}'\boldsymbol{y} \quad \text{(the normal equations)}$$

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} \quad \text{(the least squares solution)}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{w}} \quad \text{(the fitted values)}$$

Interactive demo

Exploratory data analysis

- Initial plotting/summaries of data
  - An idea about possibilities
  - Potential non-linearities, or
  - high leverage points
- Scatterplot matrix:
  - Pair-wise scatterplots
  - Histograms/density plots of single variables
- Correlation matrix:
  - Pair-wise correlations of variables (linear dependence)
  - Multicollinearity

Implementing ordinary least squares

- Instead of the mentioned least squares solution* $\hat{w} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$, we can formulate OLS with gradient descent, similar to the Adaline with:
  - Linear activation function
  - Same cost function as OLS (sum of squared errors) (the factor 1/2 introduced to make calculation of the derivative as simple as possible):

$$J(w) = \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

  - Skip Adaline's unit step function => continuous target

*Require X to be (1) full rank and (2) square -> often not the case

Model performance

- **Always try to keep your test data in the best possible condition ("mint condition")**
  - The test data must be held completely isolated from the model building (no tampering or peeking before the final validation).

- **The problem of "overfitting":**
  - "If you torture the data long enough, it will confess to anything" ~ Ronald Coase
  - Proper validation is required to assess the best possible model performance (and to avoid overfitting).

Performance metrics

- Performance metrics: evaluate the model with the goal of identifying how well the model performs on new data.
- Mean Absolute Error (MAE)
  - Consistency of error magnitude

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- Mean Squared Error (MSE)
  - Sensitive to large errors

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

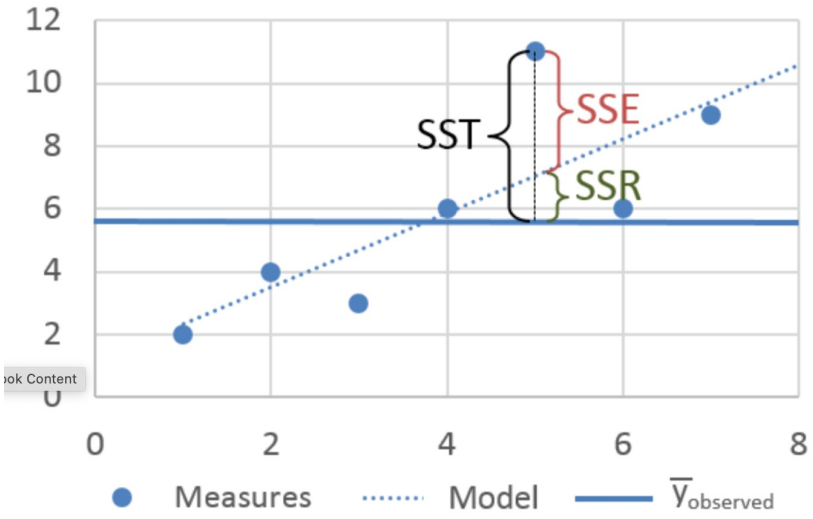
- Root Mean Squared Error (RMSE)
  - MSE, but restores the unites

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Main Notebook

R^2 - Coefficient of Determination

R-squared represents the proportion of the variance in the dependent variable which is explained by the linear regression model.



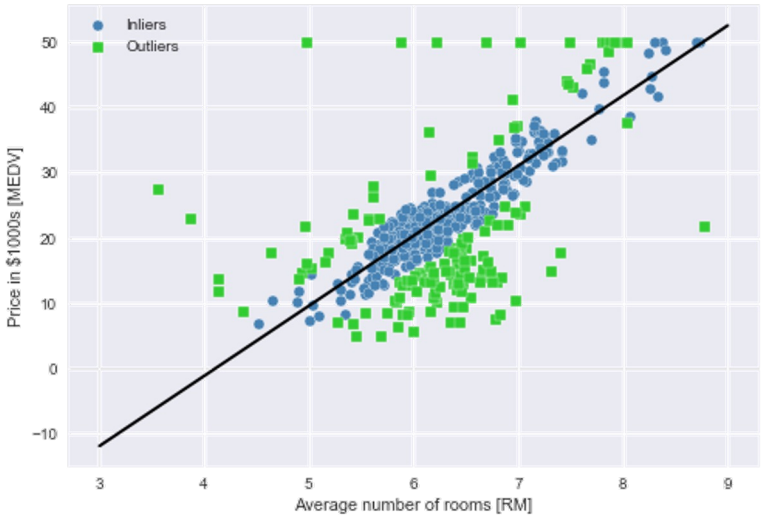
- **The Sum of Squares Total (SST)** is the squared differences between the observed dependent variable and its mean.
- **Sum of Squares Regression (SSR)** - is the sum of the differences between the predicted value and the mean of the dependent variable. (response variance)
- **Sum of Squares Error (SSE)** - the difference between the observed value and the predicted value.

$$R^2 = 1 - \frac{SSE}{SST}$$

$$= 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2}{\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \mu_y\right)^2} = 1 - \frac{MSE}{Var(y)}$$

RANSAC - RANdom Sample Consensus

- Background:
  - Estimation of **outliers** is complicated
  - **Robust** regression built on estimated inliers
  - Problem dependent thresholds



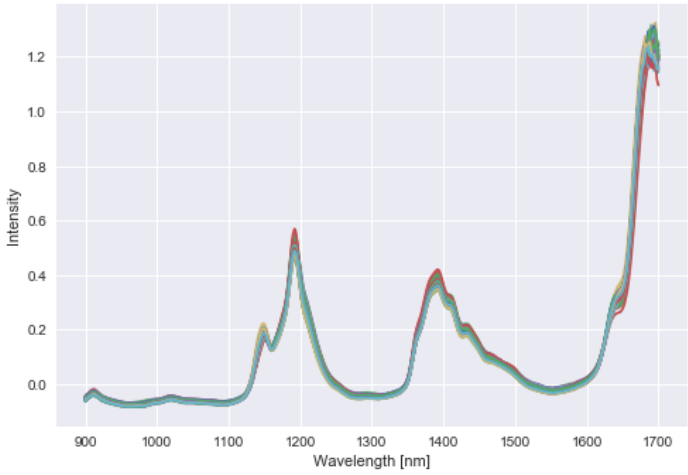
- RANSAC:
  1. Random set of inliers, fit model
  2. Test excluded samples for user-defined inlyingness, e.g., MAD based $(= median(|X_i - X_{median}|))$
  3. Refit including extra inliers
  4. Estimate error of fitted model for the inliers
  5. Terminate if performance meets user requirement or max iterations, otherwise repeat from start.
- https://en.wikipedia.org/wiki/Random_sample_consensus

Jupyter: plot_ransac.ipynb

High-dimensional data

- In spectroscopy data matrices
  - Wavelengths
- Financial data
  - ‘Unlimited’ number of factors
- Genomics
  - 3 billion DNA base pairs and their interactions

X



Curse of Dimensionality

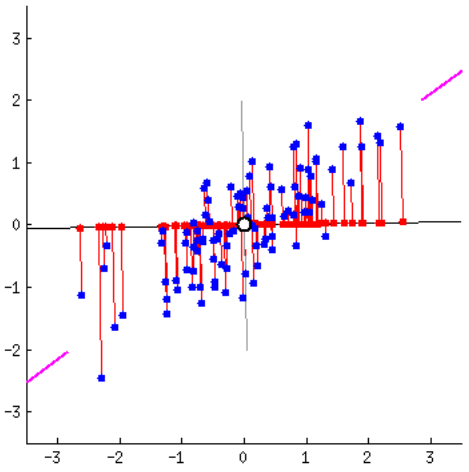
(Multi)collinearity

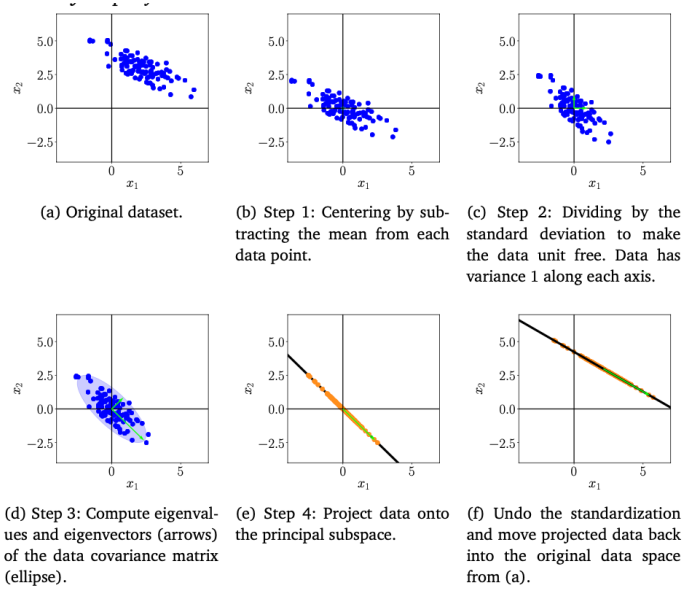
Statistical significance (less reliable/false results, R^2)

PCR (PCA Regression)

- Principle components as the regressor

1. Perform PCA on your data matrix
2. Use OLS to regress observed outcome (to get the regression coefficients)
3. (Transform the coefficient-vector back to scale (of the actual covariates))





(a) Original dataset.

(b) Step 1: Centering by subtracting the mean from each data point.

(c) Step 2: Dividing by the standard deviation to make the data unit free. Data has variance 1 along each axis.

(d) Step 3: Compute eigenvalues and eigenvectors (arrows) of the data covariance matrix (ellipse).

(e) Step 4: Project data onto the principal subspace.

(f) Undo the standardization and move projected data back into the original data space from (a).

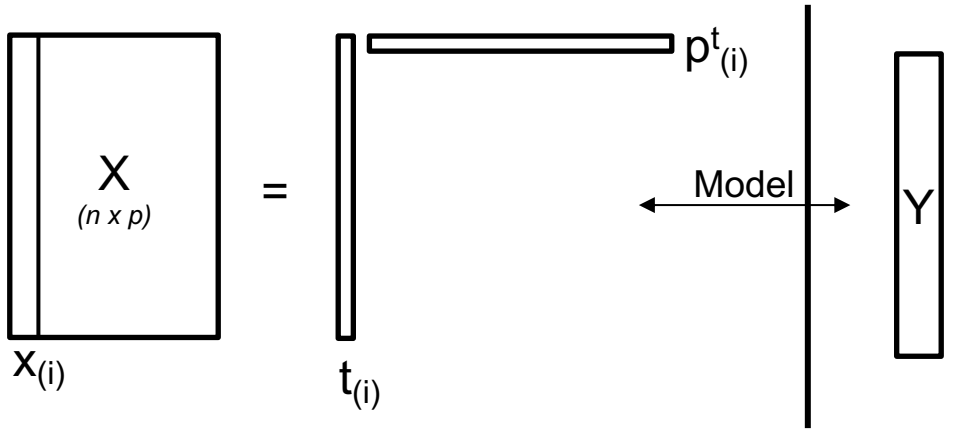
Figure 10.11 Steps of PCA. (a) Original dataset; (b) centering; (c) divide by standard deviation; (d) eigendecomposition; (e) projection; (f) mapping back to original data space.

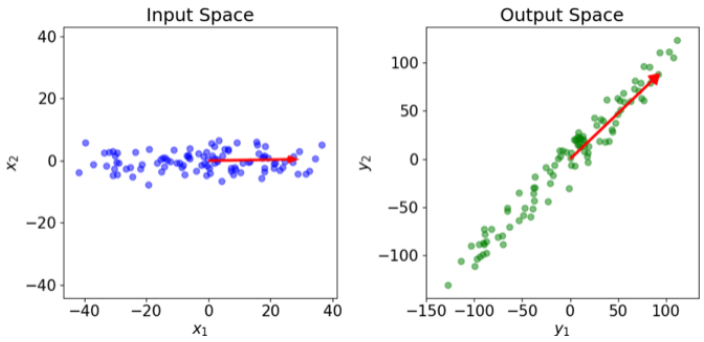
Jupyter: PCA regression

PLS* regression 🇸🇪

** **P**artial **L**east **S**quare (also called projection to latent structures)*

- (Strongly) related to PCA
  - But focus on covariance between X and Y
- Decompose matrix $X = tp^T + \varepsilon$
  - **t**: score vector
  - **p**: loading vector
- **t** should relate to Y
- **tp^T** should describe X



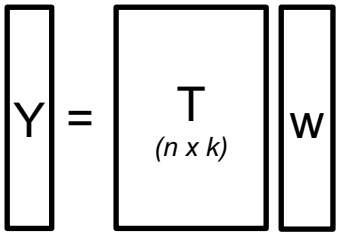


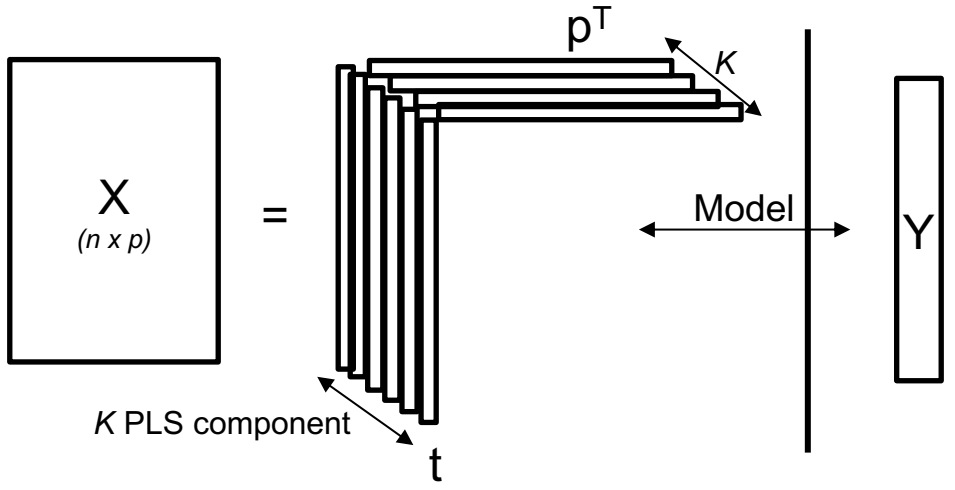
additional resource

PLS* regression

** **P**artial **L**east **S**quare (also called projection to latent structures)*

- (Strongly) related to PCA
- Decompose matrix $X = tp^T + \varepsilon$
  - Repeat *K* times to extract PLS components
  - Each components are orthogonal to each other
- Regression on score matrix





additional resource

Jupyter: pls_regression.ipynb

PCR vs PSL

	PCR	PLS
Advantages	**Dimensionality** reduction - effective at reducing the dimensionality of data	**Covariance** - between the predictors and the response is targeted
	Multicollinearity – By using PC's the problems of multicollinearity is reduced	**Multicollinearity** – also handles multicollinearity well
	Interpretation - The PC regressors can (sometimes) be interpreted	**Small data sets** - PLS can work effectively even with a smaller number of observations
Disadvantages	**Variance-Capture** - capturing the variance in the predictors, not the variance that is most predictive of the response variable.	**Complexity** - computationally intensive and possibly harder to implement or interpret
	Information Loss - By ignoring smaller principal components	**Interpretability** - components are a combination of directions that explain both the predictors and the response.
	Response Variable - PCR does not consider the response variable when determining the PCs	**Overfitting Risk** - maximize covariance risks overfitting

Ease of use > accuracy
predictive space > predictive model

Transformations in regression

- Non-linear patterns in “residuals vs fitted values”, “response vs features”, “scatter plot matrix”, …
  - Transform predictors or target
  - Polynomials, log, Box-Cox (scipy.stats), …
  - Beware of overfitting!

- Predictor transformation:

$$y = w_0 + w_1 x + w_2 x^2 + \ldots + w_d x^d$$

Beaver of overfitting

- Target transformation: (assuming a relationship)

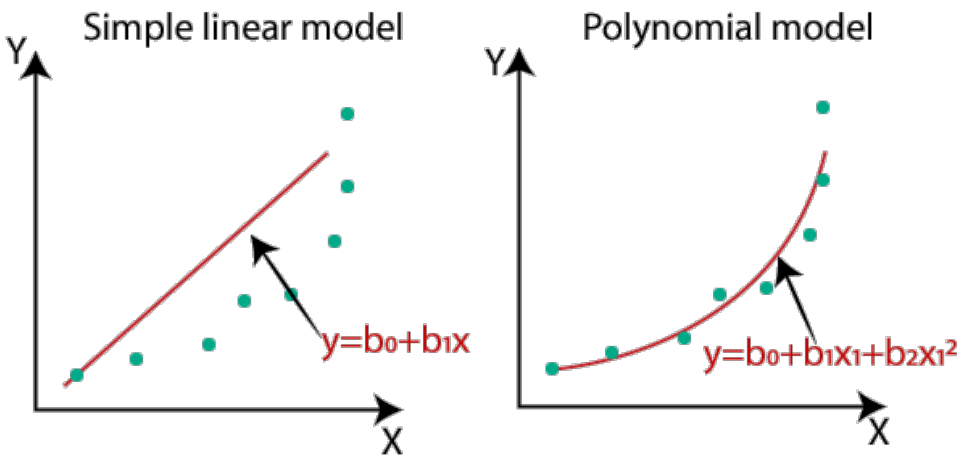
$$f(x) = 2^{-x}$$

$$\log(f(x)) = -x$$

Polynomial regression

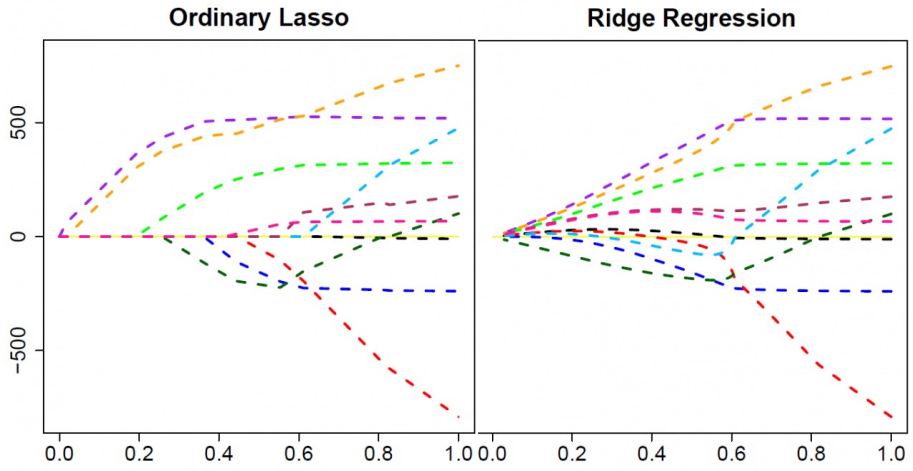
- Non-linear data, but linear coefficients
- N-th degree of polynomial
  - $X, X^2, X^3, \ldots, X^n$
- $\hat{y} = \hat{w}_0 + \hat{w}_1 x + \hat{w}_2 x^2 + \cdots + \hat{w}_p x^n$

Simple Linear Regression	$y = b_0 + b_1 x_1$
Multiple Linear Regression	$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$
Polynomial Linear Regression	$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$



Jupyter: polynominal_regression.ipynb

Regularization



- OLS: $J(w)_{OLS} = \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right)^2$

- Ridge: $J(w)_{Ridge} = \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 + \lambda \| w \|_2^2$ $\quad L2: \quad \lambda \| w \|_2^2 = \lambda \sum_{j=1}^{m} w_j^{\,2}$

- Lasso: $J(w)_{LASSO} = \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 + \lambda \| w \|_1$ $\quad L1: \lambda \| w \|_1 = \lambda \sum_{j=1}^{m} \left| w_j \right|$

- Elastic net: $J(w)_{ElasticNet} = \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 + \lambda_1 \sum_{j=1}^{m} w_j^{\,2} + \lambda_2 \sum_{j=1}^{m} \left| w_j \right|$

(can be reformulated as a linear SVM)

Regularization

L2:

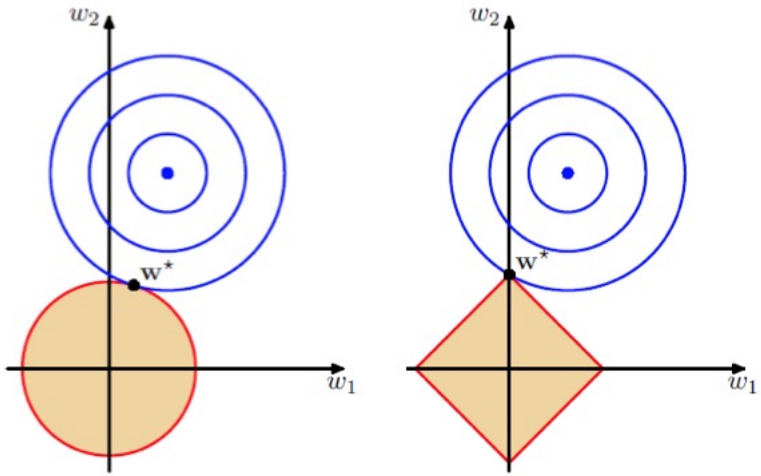
- **Mathematically:** always invertible, eigenvalues away from zero

$$\beta = (X^T X + \lambda I)^{-1} X^T Y$$

- **Geometrically:** (hyper-) sphere constrains
- **Conceptual:** shrinking of the coefficients

L1:

- **Mathematically:** absolute value
- **Geometrically:** polytope constraints
- **Conceptual:** sparsity of the coefficients



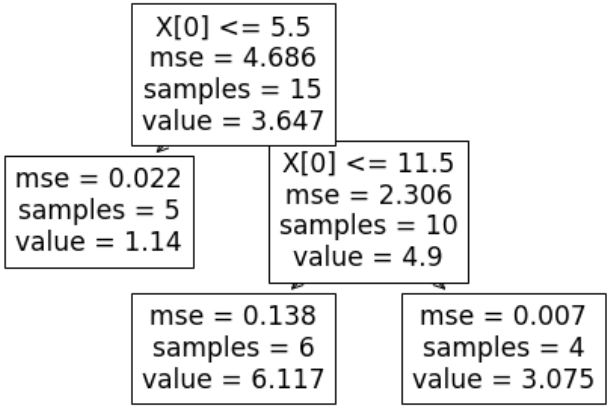
L2
(Ridge)

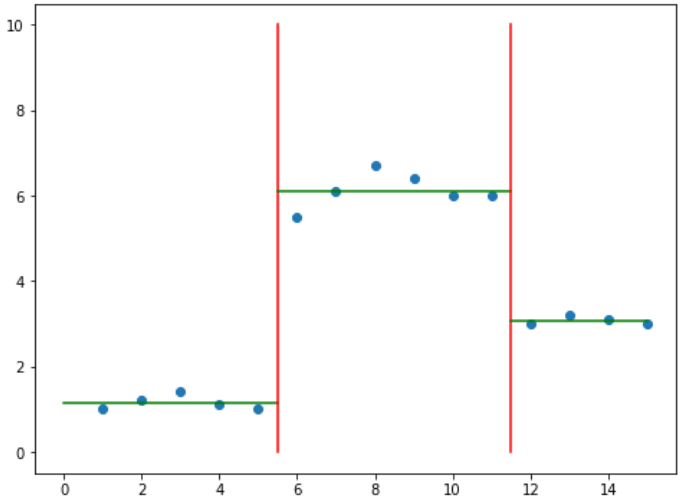
L1
(Lasso)

Jupyter: regularization.ipynb

Decision tree regression

- Local, piecewise linear modelling
  - Instead of global modelling
  - Can approximate most non-linear relationships without transformation
- Decision tree feature splitting by **Information Gain** (choosing which feature, and which value to split at):
  - Classification: Gini impurity
  - Regression: MSE



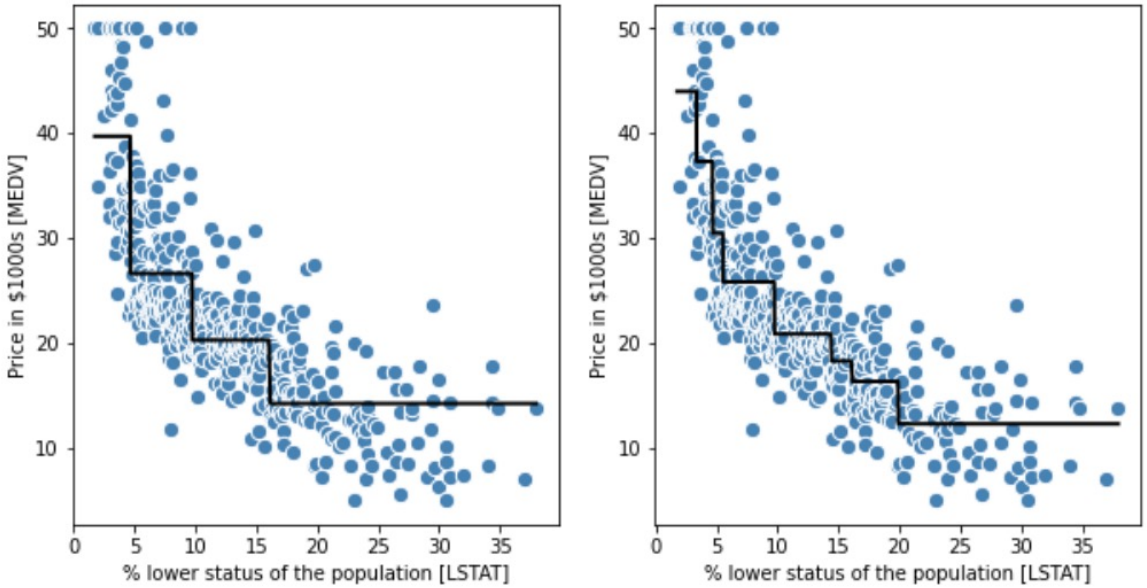


Jupyter: decision_tree.ipynb

Random forest regression

- Generalisation of decision trees corresponding to classification:
  - Use MSE for tree growing
  - Predict by average over all trees
  - Prone to overfitting

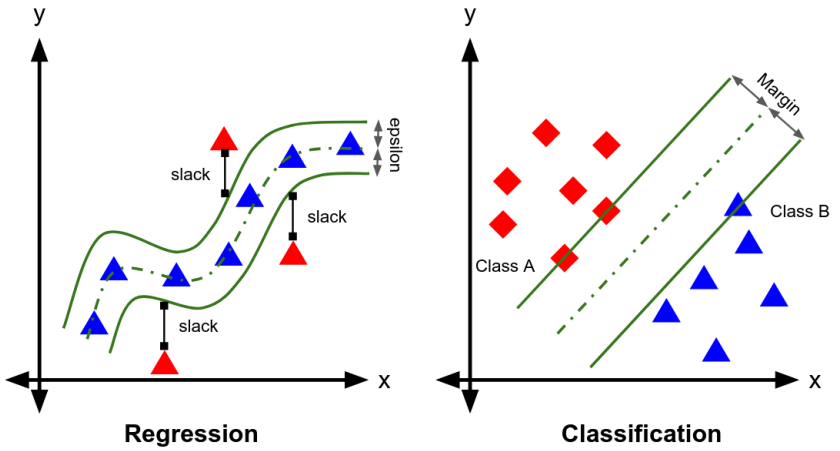
- RF samples both objects and features

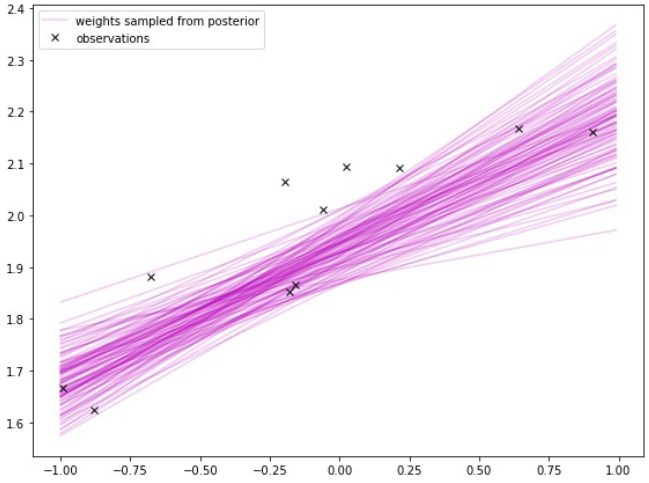


Jupyter: Chapter 10, part 2

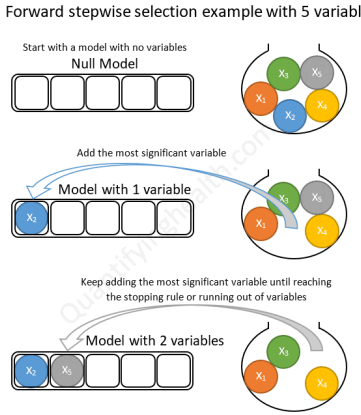
Other types of Regression

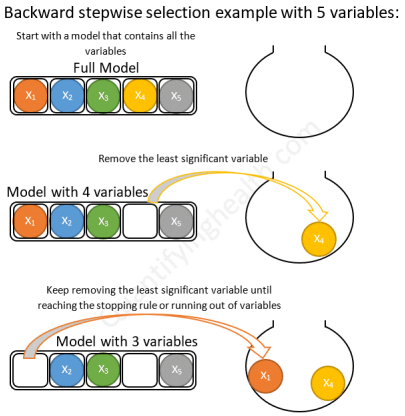
- Support Vector Regression (SVR)
- Stepwise Regression
  - Forwards and Backwards
- Bayesian Linear regression





$$p(y|\boldsymbol{X}, \beta, \sigma^2)$$





Regression as the last step of …

- Several steps, e.g.:
  - feature transformation,
  - feature engineering,
  - data compression, and/or
  - feature selection, before
  - regression
  - (or classification)

- NOTE: Many of the above steps can be pipelined…

Summary

Beaver of overfitting

- Exploratory data analysis:
  - Plot and summarize before analysis
  - Errors, relationships, potentials
- OLS can be formulated as GD
- Regression still works if #features >> #objects (if treated properly)
- Outlier robustness through RANSAC
- Residuals vs fitted values
  - Model assumptions
  - Patterns leaking into residuals
  - Transformations
- RMSE as alternative to MSE
- Tree based regression