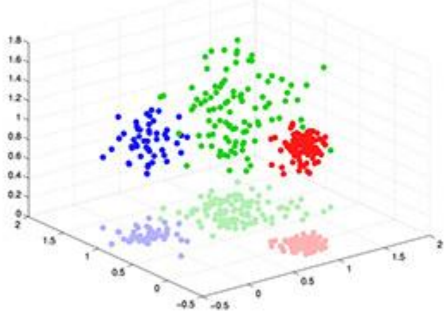
MACHINE LEARNING FOR REGRESSION

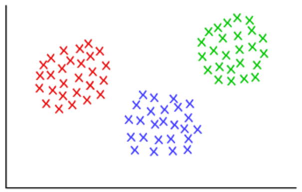
Curriculum

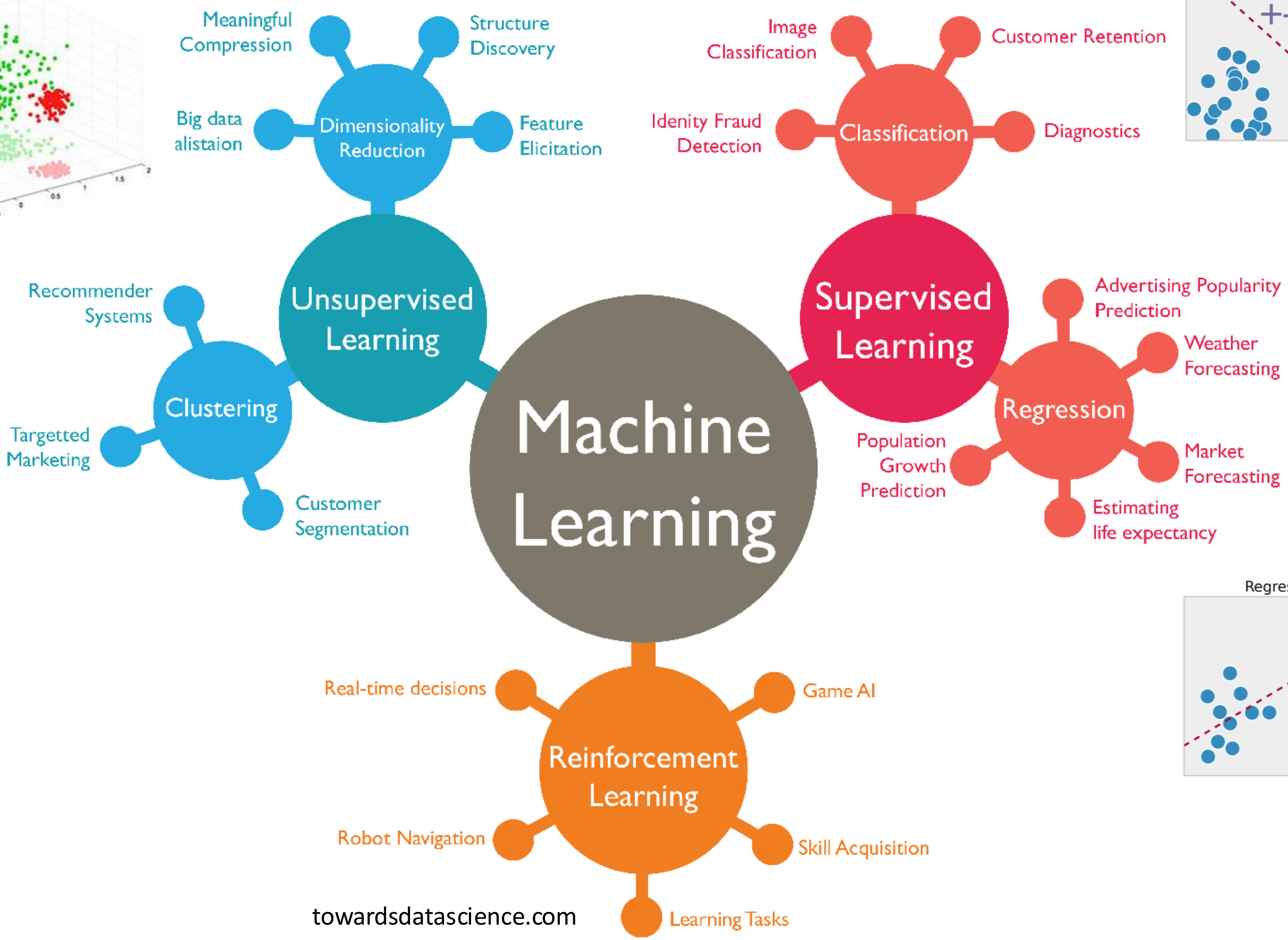
- What is a regression task?
- How does linear regression work?
  - Fitting a univariate linear function to your data
  - From univariate, to multivariate, to non-linear regression
  - Optimizing loss functions
- How do I evaluate the quality of a regressor?
- Regularized regression

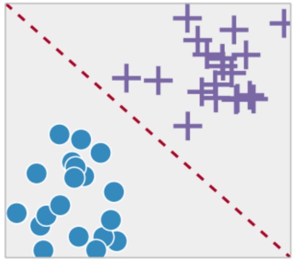
What is a regression task?

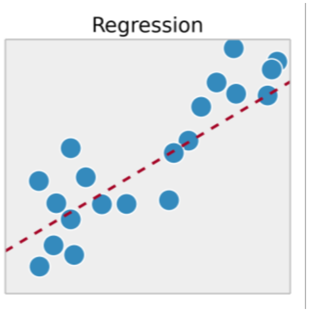
SINTEF











towardsdatascience.com

SINTEF

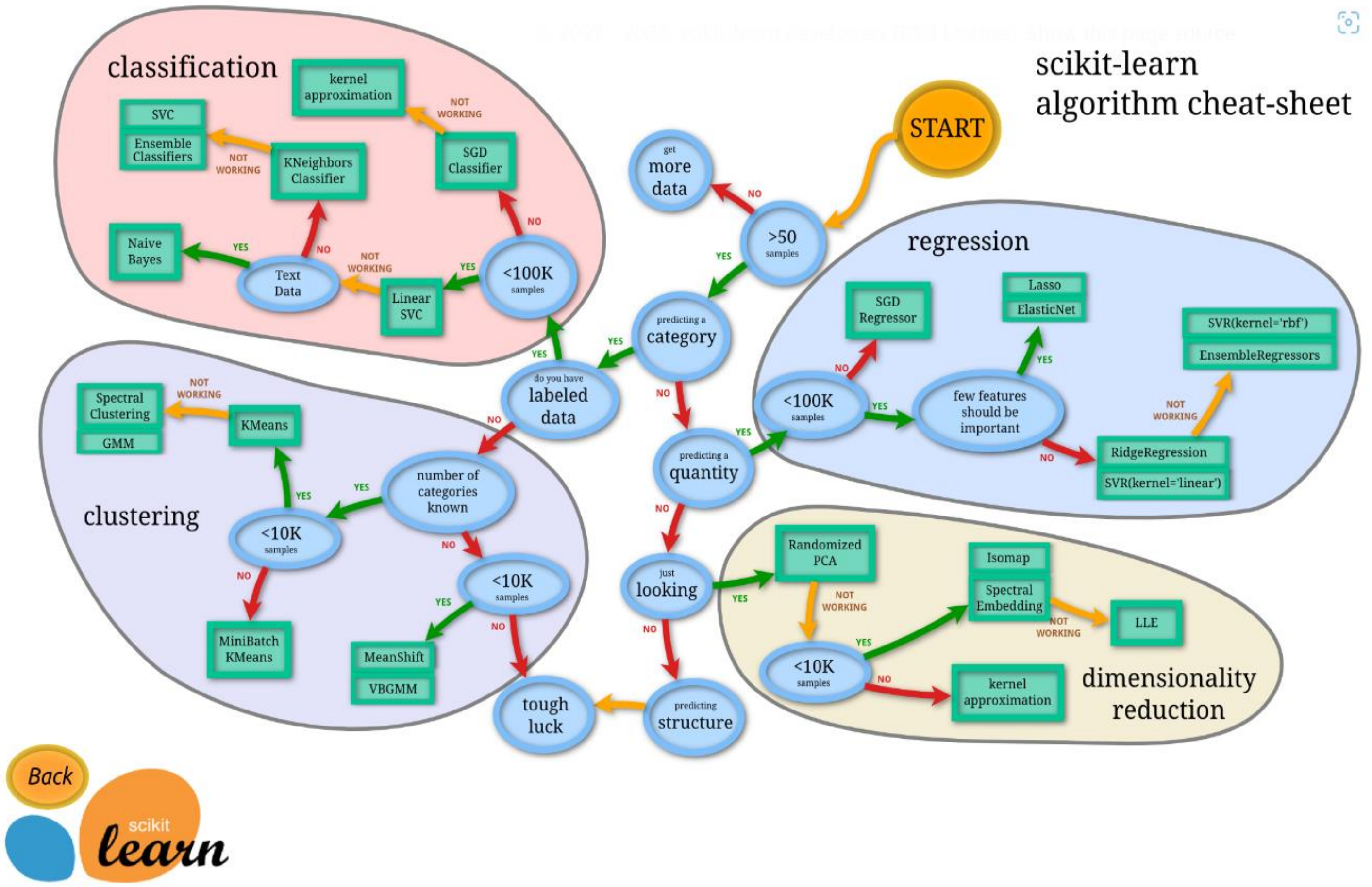
What is a regression task?

- Tasks: Find a function (aka regressor) that models the relationship between an input and an output
  - Input – A set of independent variable X – Also known as features
  - Output – A scalar value y – Also known as target/dependent variable
- Examples:
  - Predicting house prices given their features: Size in square meters, post code, number of floors, has garden?
  - Predicting sales numbers, travel time, electricity consumption, ...

SINTEF

(The art of) Choosing the right approaches

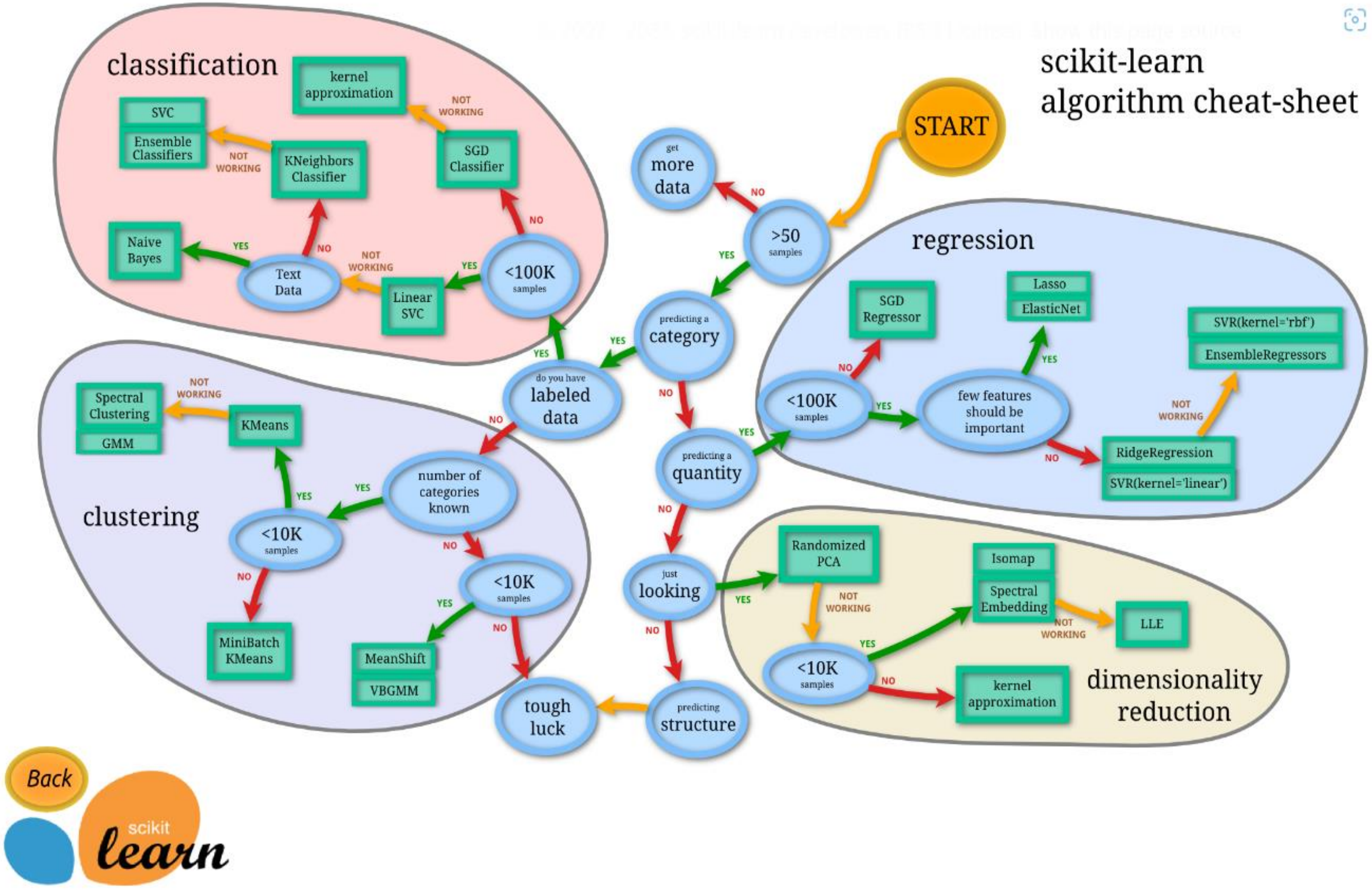
- In general difficult to know what will work best
- Useful guidelines:
  https://scikit-learn.org/stable/machine_learning_map.html



SINTEF

Quiz -

- Use-cases:
  - A - Predicting the price of a house given a dataset of 50k sales acts and house features
  - B - Predicting stock prices based on historical data, market indicators, and other economic factors with 1+ million points

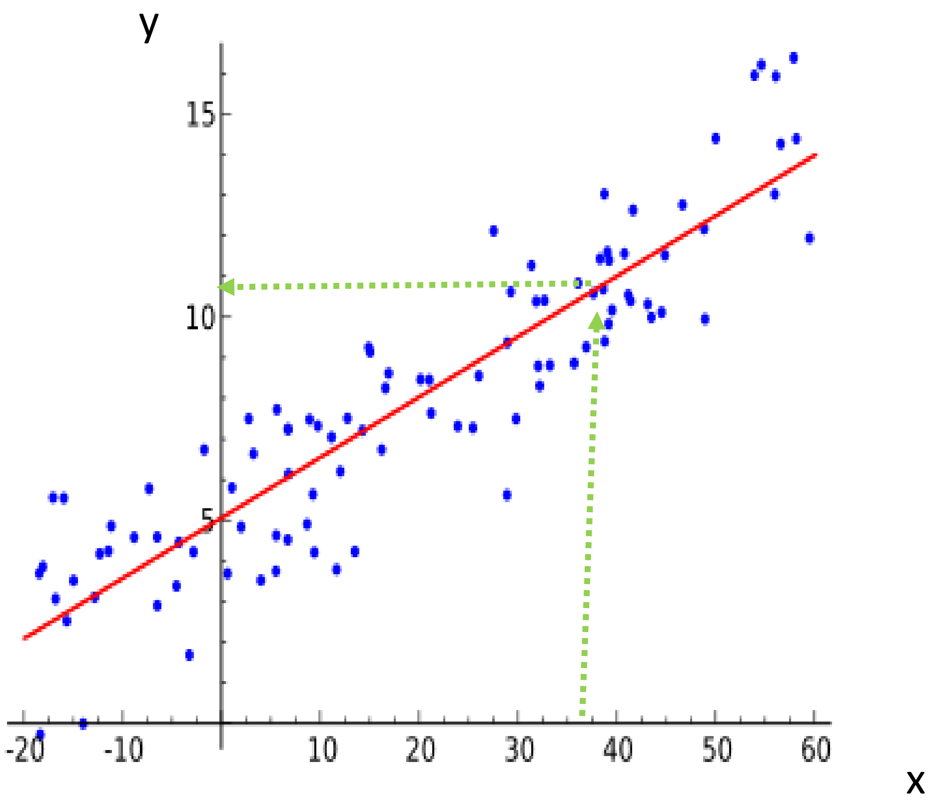


SINTEF

How does linear regression work?

Regression –
Fiting function parameters to data

- We want to model the relationship between a variable X and a variable y
  - Once done, we can use that function to predict y of an unseen data point (x)



SINTEF

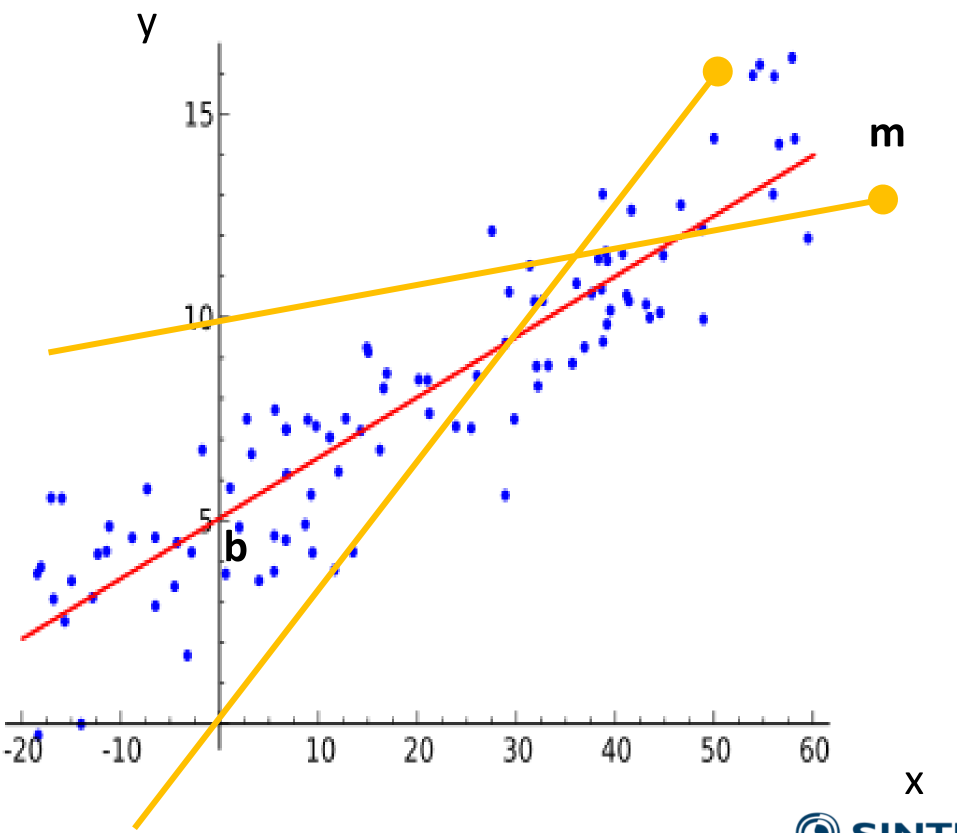
Regression –
Fitting a linear function

- Let us use a very simple function:
  - y = **m**X + **b**
  - **m** and **b** are the parameter of the model
- But what are good values for **m** and **b**?
  - We want to find **m** and **b** so to minimize a distance between the line and the data.

$$Cost\ Function(MSE) = \frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\ pred})^2$$

Replace $y_{i\ pred}$ with $mx_i + c$

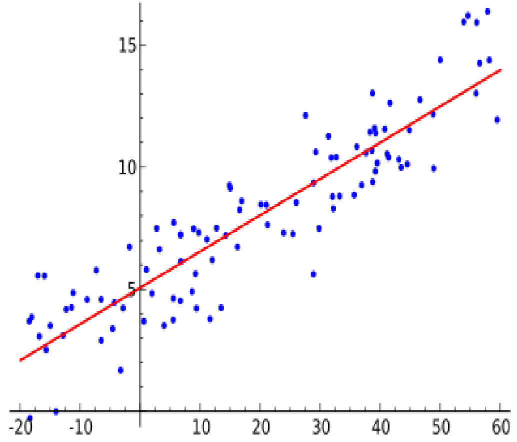
$$Cost\ Function(MSE) = \frac{1}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))^2$$

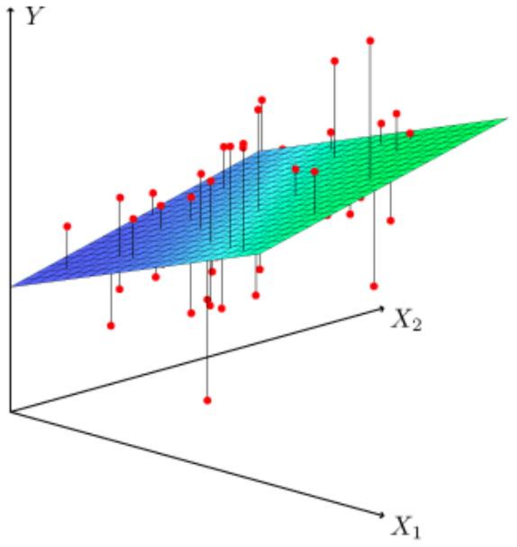


SINTEF

Regression – From univariate to multivariate linear regression models

- Univariate
  - y = **m**X + **b** (m, b are scalars e.g. m=0.6, b=5.1)
- Multi-variate
  - With 2 dependent variable we describe a plane in 3d space
  - With more than **n** dependent variable, we descibe a n-dimensional hyper-plane in **n+1** hyper-space.
  - y = **m_0** * X_0 + **m_1** * X_1 + … + **m_n** * X_2 + **b**
    - y = **m**X + **b** (m is an n-dimentional vector m = (m_0, m_1, …, m_n))



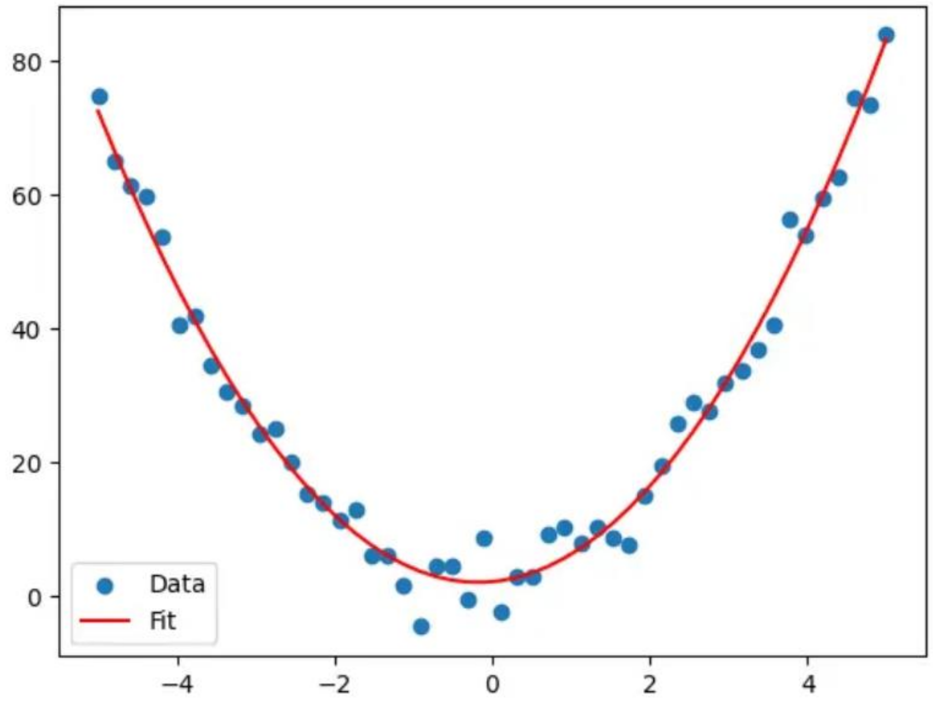


x

SINTEF

Regression – From Linear models to non-linear models

- Linear function may not be sufficient to model your data!
- … But the idea remains the same
  - Find model parameters that minimize a loss function

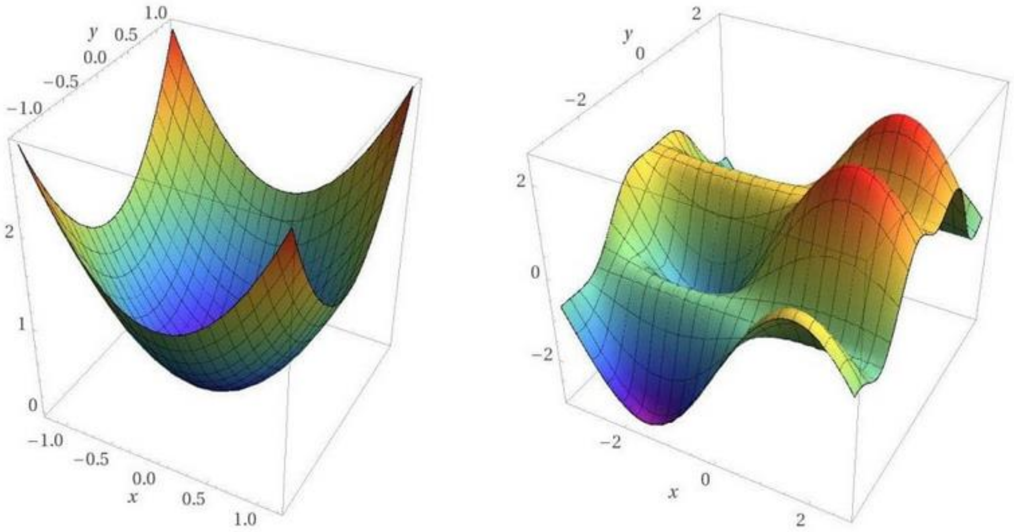


x

SINTEF

Regression – Minimizing the loss function

- Depending on your model choice and your choice of loss function
  - Finding a minimum can be "easy"
    - Algebraic tricks
    - Linear solvers
  - Or more involved ...
    - Specialized solvers
    - Gradient Descent – A swiss knive to find minima



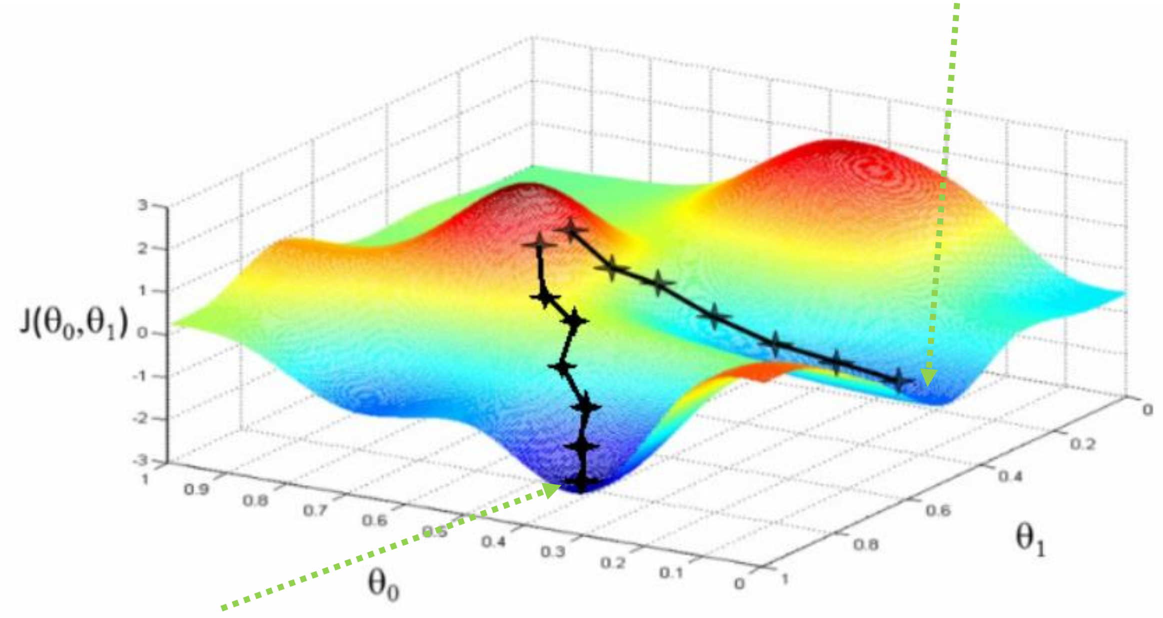
x

SINTEF

Regression – Optimizing your loss functions via Gradient descent

- An algorithm used to find the minimum of an arbitrary (loss) function
  - Select a random starting point
  - Look around you
  - Take a step in the steepest downhill direction.
  - Repeat :)
- No guarantee to end up in a global minimum!

Local minimum

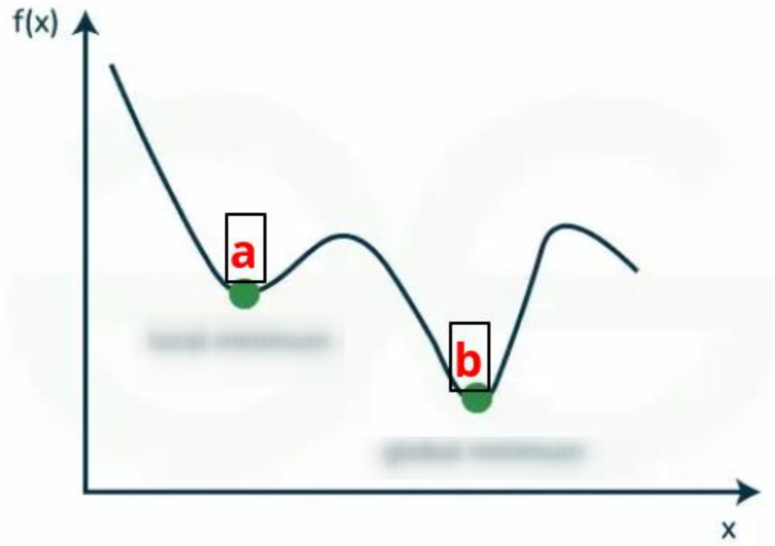


Global minimum

SINTEF

Quizz

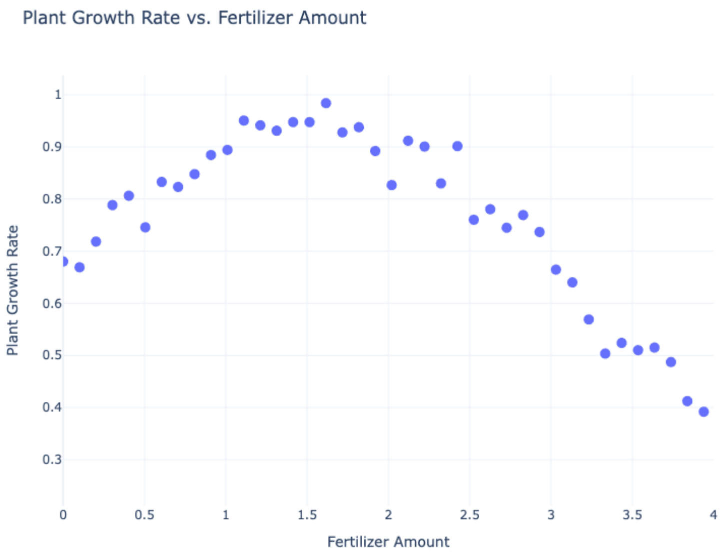
- Spot the global minimum
  - A?
  - B?



SINTEF

Regression – What model should I choose?

- Modeling linear relationships
  - **1 feature?** Univariate linear regression
  - **More features?** Multivariate linear regression
- Modeling non-linear relationships
  - Polynomial regression, exponential, logarithmic regression



- Need to model arbitrarily complex relationships?
  - Neural networks

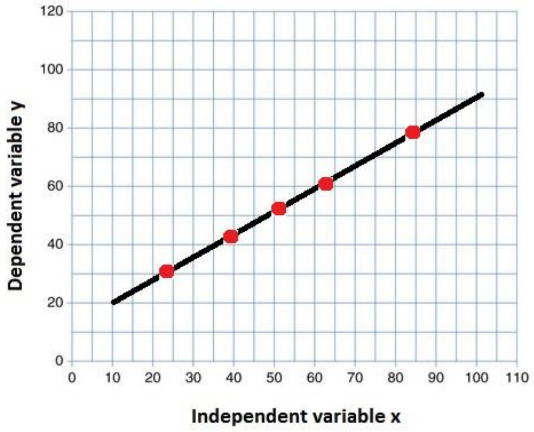
- **Model complexity:**
  - Complex models can be more prone to overfitting
  - Hard to interpret, difficult to use
- **Generalizability:**
  - Ensure the model generalizes well to new data.
- **Domain knowledge:**
  - Consider domain knowledge when interpreting model results

SINTEF

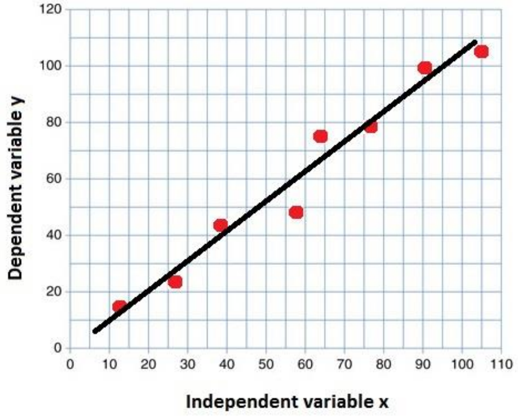
How do I evaluate a regressor?

Measuring the quality of a regressor - (Linear regression)

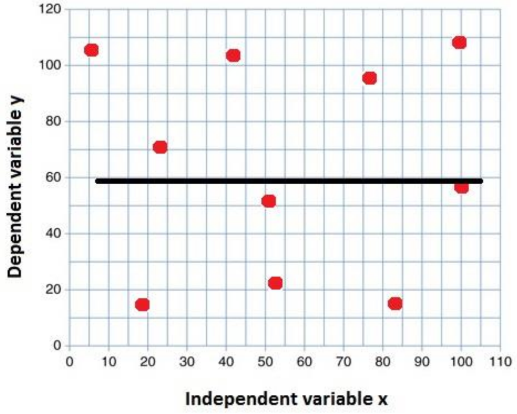
- Coefficient of determination (**RSquared)**
  - Measures the proportion of variance in the dependent variable that is explained by the independent variables
  - Provides an overall measure of how well the model fits the data
  - Only used for linear regression



Rsquared = 1.0



Rsquared = 0.85



Rsquared = 0.0

SINTEF

Measuring the quality of a regressor - Common metrics

- Mean Absolute Error (**MAE**):
  - The average (absolute) error between predicted and actual value.
    - No differentiation between one huge error and many small ones.

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

- Mean Squared error **(MSE)**
  - Very commonly used – Friendly to optimize.
  - Use when you want to avoid large errors (and prefer a bunch of smaller ones)

$$MSE = \frac{\sum(y_i - \hat{y}_i)^2}{n}$$

- Root Mean Squared Error (**RMSE**):
  - Punishes larger errors more than MAE

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

https://www.kaggle.com/code/marcinrutecki/regression-models-evaluation-metrics/notebook

Measuring the quality of a regressor - Defining a loss function

- Many more evaluation metrics...
- It sometimes makes sense to write your own.
- Context, goal and application dependent!

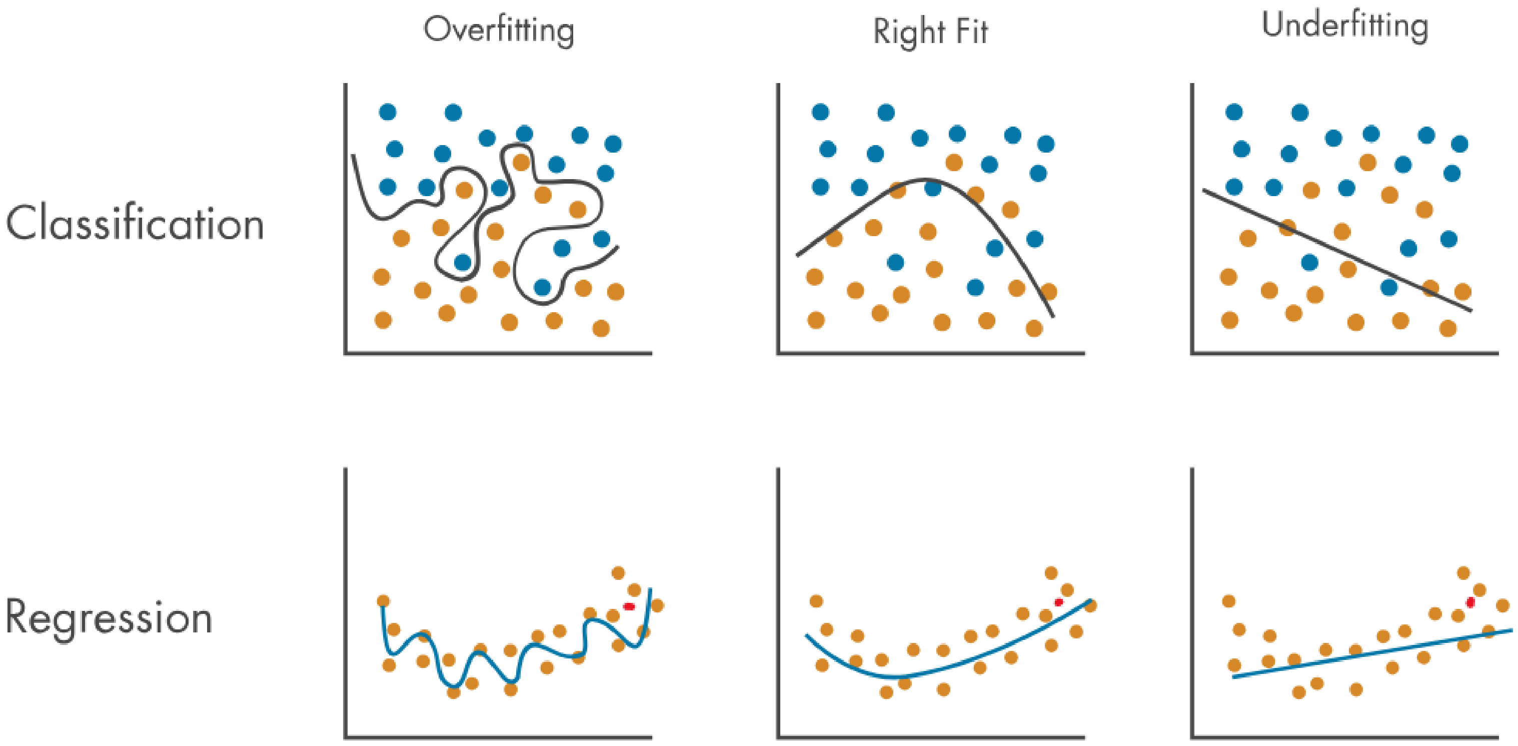
- Example of own metric.
  - My business can predict house prices with an error of RMSE 10.25601
  - My business predicts the price within 10% error margins, 90% of the time (Own business centric metric)

Quizz - Can you find the best model?

The importance of regularization

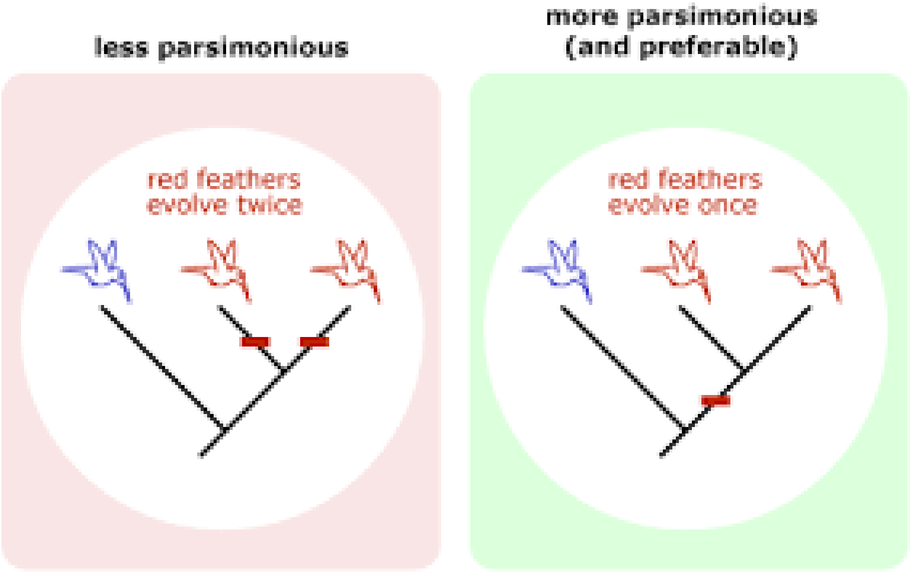
SINTEF

The problem of model overfitting



- -> Good performance on previously seen data points but poor performance on new data
- https://www.mathworks.com/discovery/overfitting.html

Regularization as a fix



- Adding a penalty term to the loss function
- This penalty term makes the model prefer simpler solutions - Principle of parsimony
  - Gain: Potential increased performance on the testing data (new data points)
  - Loss: Potential decreased performance on the training data (known data points)

SINTEF

https://undsci.berkeley.edu/understanding-science-101/how-science-works/competing-ideas-other-considerations/

Regularization as a fix

- Loss function without regularization :

$$L = \underbrace{\boxed{\sum(y-\hat{y})^2}}_{\text{Sum of Squared Residuals}}$$

- **Ridge regularization** adds the sum of squared coefficients to the loss function ->

$$L = \underbrace{\boxed{\sum(y-\hat{y})^2}}_{\text{Sum of Squared Residuals}} + \underbrace{\boxed{\alpha\sum m^2}}_{\text{Penalty}}$$

- **Lasso regularization** adds the sum of absolute values of coefficients->

$$L = \underbrace{\boxed{\sum(y-\hat{y})^2}}_{\text{Sum of Squared Residuals}} + \underbrace{\boxed{\alpha\sum |m|}}_{\text{Penalty}}$$

- **Elastic Net Regularization** adds both ->

$$L = \underbrace{\boxed{\sum(y-\hat{y})^2}}_{\text{Sum of Squared Residuals}} + \overbrace{\underbrace{\boxed{\alpha\sum m^2}}_{\text{Penalty}}}^{\text{RIDGE}} + \overbrace{\underbrace{\boxed{\alpha\sum |m|}}_{\text{Penalty}}}^{\text{LASSO}}$$

https://medium.com/@tavishi.1402/regularization-techniques-lasso-and-ridge-90d3cc73ca4c

Using regularization techniques

- Give you control over model complexity
  - Complex enough to capture underlying patters
  - Not complex enough to fit noise and random fluctuations in the dataset.

- Ridge is **effective with highly correlated features**
- Lasso is **effective with a dataset with many features**
- Elastic net give you the best of both worlds :)

SINTEF

https://medium.com/@tavishi.1402/regularization-techniques-lasso-and-ridge-90d3cc73ca4c

SINTEF

Technology for a better society