# Regularisation

# Table of Contents

Concepts Review

Regularisation

- L2 - Ridge
- L1 - LASSO

Information Criterion

- Akaike Information Criterion (AIC)
- Bayesian Information Criterion (BIC)
- Recursive Feature Elimination (RFE)

Data Science process

## Learning Goals I

At the end of this lecture you should be able to:

- Understand what **Regularisation** means
- Understand how **Ridge - L2** regularisation works
- **Fit** a Ridge regression model with sklearn
- Understand how **LASSO - L1** regularisation works
- **Fit** a LASSO - L1 regression model with sklearn
- Understand Regularisation **requirements** and know when to use them
- **Scale** the inputs of a regularisation model

## Learning Goals II

At the end of this lecture you should be able to:

- Understand what is an **Information Criterion** and when to use it
- Understand the difference between **AIC** and **BIC**
- **Calculate** AIC and BIC with sklearn
-
-

## Concepts Review

## Concepts review

How do you prepare your data for Polynomial Regression?

How do you fit Polynomial Regression with sklearn?

What’s the graphical intuition of MSE?

How does MSE decompose? What’s the graphical intuition of its components?

## Regularisation

## Regularisation

Adding a **complexity Penalty** to the Cost function

This penalty is proportional to the **size** of the **coefficients**

As a result learners will try to avoid models with **large coefficients** that don’t overcompensate with performance

It allows us to **reduce** variance while **increasing** bias

## Regularisation Assumptions

When penalising, regularisation assumes all inputs in the same scale

We need to **scale** all inputs before we fit the model!

## L2 Regularisation - Ridge Regression

Penalty proportional to the square of the coefficients

Prioritises reducing the largest coefficients first (Ceteris Paribus)

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

Cost function for ridge regression

## Ridge Regression (10 steps)

1. Get your data in a numpy or pandas format
2. Separate your **predictors** from your **target** variable (X,y)
3. Import StandardScaler (from sklearn.preprocessing)
4. Instantiate StandardScaler
5. Scale your X
6. Import the **Ridge** function (from sklearn.linear_model)
7. Create an instance of Ridge (you can pass alpha at this point)
8. Pass X, y to train your model
9. Have a look at the coefficients and intercept
10. Validate on your test dataset

DEMO

## Your turn to practice Ridge Regression
## Are you ready?!

## L1 Regularisation - LASSO

Least Absolute Shrinkage and Selection Operator (LASSO)

Penalty proportional to the absolute value of the coefficients (no negatives)

Prioritises zeroing coefficients of inputs with the least value first (Ceteris Paribus)

Can be used as an embedded technique for Feature Selection

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

Cost function for Lasso regression

# LASSO Regression (10 steps)

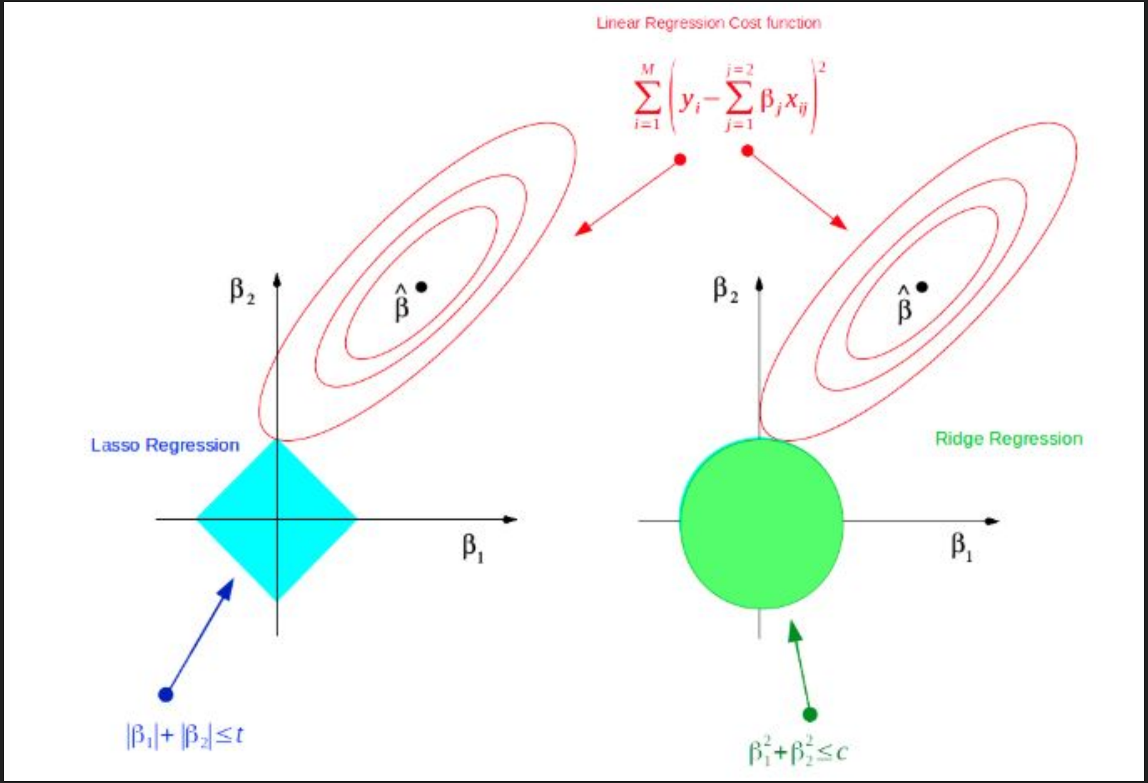
1. Get your data in a numpy or pandas format
2. Separate your **predictors** from your **target** variable (X,y)
3. Import StandardScaler (from sklearn.preprocessing)
4. Instantiate StandardScaler
5. Scale your X
6. Import the **Lasso** function (from sklearn.linear_model)
7. Create an instance of Lasso (you can pass alpha at this point)
8. Pass X, y to train your model
9. Have a look at the coefficients and intercept
10. Validate on your test dataset

DEMO

Your turn to practice LASSO Regression

Are you ready?!

# Visual interpretation on Regularisation


Linear Regression Cost function
$\sum_{i=1}^{M}\left(y_i - \sum_{j=1}^{j=2}\beta_j x_{ij}\right)^2$
$\beta_2$
$\hat{\beta}$
$\beta_1$
Lasso Regression
$|\beta_1| + |\beta_2| \leq t$
$\beta_2$
$\hat{\beta}$
$\beta_1$
Ridge Regression
$\beta_1^2 + \beta_2^2 \leq c$

Explain to your partner Regularisation

## Information Criterion

## Information Criterion

The model with the lowest IC should be selected

Feature selection

## Akaike Information Criterion

The complexity penalty is on the scale of the likelihood

$$\mathrm{AIC} = 2k - 2\ln(\hat{L})$$

## Bayes Information Criterion

The complexity penalty is on log scale

$$\mathrm{BIC} = \ln(n)k - 2\ln(\widehat{L}).$$

## Information Criteria (10 steps)

1. Get your data in a numpy or pandas format
2. Separate your **predictors** from your **target** variable (X,y)
3. Import StandardScaler (from sklearn.preprocessing)
4. Instantiate StandardScaler
5. Scale your X using the .fit_transform() in your StandardScaler object
6. Import the LassoLarsIC function (from sklearn.linear_model)
7. Create an instance of LassoLarsIC (select ‘aic’ or ‘bic’ for criterion)
8. Pass X, y to train your model
9. Have a look at the coefficients and intercept
10. Validate on your test dataset

DEMO

Your turn with IC
Are you ready?!

## Reflection

What have you learned?

What outcomes do you still feel you need to work on?

What steps are you going to take to make that happen?