Ridge & Lasso Regression & Elastic Net

Ridge Regression

→ Cost function in case of Ridge Regression.

$$CF = \underbrace{\sum_{i=1}^{n} \left(y_i - \sum_j x_{ij} \beta_j\right)^2}_{\text{From Linear Regression.}} + \underbrace{\lambda \sum_{j=1}^{p} \beta_j^2}_{L_2 \text{ Regularization.}}$$

→ This reduces variance in Linear Regression Model.
→ This increases bias in Linear Regression Model.
→ This prevents overfitting.
→ λ learned through cross-validation / hyperparameter tuning.
→ This reduces model complexity & over-fitting.

→ This puts contraints on the coefficients (w)
→ λ (penalty term) reduces overall slope
→ So, Ridge Regression shrinks coefficients and helps to reduce model complexity & multi-colinearity.

Lasso Regression.

→ Cost function in case of Lasso Regression.

$$CF = \underbrace{\sum_{i=1}^{n} \left(y_i - \sum_j x_{ij} \beta_j\right)^2}_{\text{From Linear Regression}} + \underbrace{\lambda \sum_{j=1}^{p} |\beta_j|}_{L_1 \text{ Regularization.}}$$

→ Reduces Variance, Increases bias, Reduces overfitting & complexi[ty]
→ λ (penalty term) reduces overall slope and is learnt through cross-validation / hyperparameter tuning.
→ constraints β
→ Reduces multi-collinearity

→ Helps in feature selection as reduces slope of coeffs (of some) features to 0 (zero).

ElasticNet

→ When there are tons of variables

→ Cost Function:

$$Cf = \underbrace{\sum_{i=1}^{n} \left(y_i - \sum_j x_{ij} \cdot \beta_j\right)^2}_{\text{Least Squares}} + \underbrace{\lambda_1 \sum_{j=1}^{p} \beta_j^2}_{L2} + \underbrace{\lambda_2 \sum_{j=1}^{p} |\beta_j|}_{L1}$$

→ There may or may not be mixing in L1 & L2

→ $\lambda_1 = 0$, Lasso & $\lambda_2 = 0$, Ridge

→ Strengths of both Ridge & Lasso.

→ Very good at dealing with situations when there are correlations b/w parameters.

→ Lasso picks only 1 correlated feature while Ridge tends to shrink all parameters for correlated variables together.

NOTE:

$\lambda, \lambda_1, \lambda_2$ all greater than 0.