Week 3: Model Selection and Model Regularisation

Overview:

- Model selection: introduction
- Elements of statistical decision theory:
  - Loss measures
  - Optimal decision for classification
  - In-sample vs out of sample loss
- Model selection criteria
- Cross Validation

Model selection:

Model or Method: Key Considerations

Functional Form

- **Linear vs Non-linear:** Models can assume a **linear relationship** (**e.g., Y=XB**) or **non-linear relationships**.
  - Some models combine both, *like Generalized Additive Models (GAMs).*
- **Variable Transformations**: Sometimes it helps to transform variables, e.g., taking logarithms: ***e.g., $Y = X^B = LnY = LnXB$***
  - Transformations can improve model fit and interpretability.

Number of Variables

- **Trade-off:** More variables can capture more complexity but may lead to overfitting.
- **Parsimony Principle**: Aim to minimize the number of parameters while retaining predictive power — simpler models often generalize better.

Variable Selection

- Decide which predictors to include. Including irrelevant variables increases variance; omitting important ones introduces bias.

How many observations to include:

- In time-series if there are structural breaks, we may want to limit the sample size.
- If there are outliers, we may want to eliminate or down-weight them!

How many observations to include when you do forecasting?

- Avoid **structural breaks** (changes in the underlying data process) that invalidate inferences.
- Ensure sufficient sample size: **n>>p**
  - (*observations much larger than number of parameters*) for reliable estimation.
- **Windowing Techniques:**
  1. **Expanding window:** Use all data up to time t to hence estimate the red point.
  2. **Rolling window:** Use the most recent k observations to account for changing dynamics to hence predict the red point.

Combination of Models

Bias-variance trade-off: **complex models** (many parameters) tend to have low bias but high variance, **simpler models** often have higher bias but lower variance.

- **Model Averaging: Idea:** Control variance within each model (use simple model) and mitigate specification bias by averaging across diverse models!
- **Weighted combinations of models** often improve forecasts:

$$\hat{y} = \sum_{m=1}^{M} w_m, \sum_{m=1}^{M} w_m = 1$$

➔ That is combine predictions from multiple models (y^m) using weights wm.

- NOTE: Choosing the weights w can be tricky — sometimes **simple averaging works well** ("hard to beat 1/M")

Elements of statistical decision theory:

Classic Regression: Standard goodness of fit measures

Sum of Squares:

- **Total Sum of Squares**: Total variation of the observed values of y around their mean — how spread out the data are overall.

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

- **Residual Sum of Squares:** The part of the variation not explained by the model — squared prediction errors.

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

- **Explained Sum of Squares**: the part of the variation explained by the regression — improvement over just using the mean.

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

- **Sum of Squares:**

$$TSS = ESS + RSS$$

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

$$TSS = how\ much\ variation\ is\ explained\ by\ the\ predicted\ values + how\ much\ variation\ is\ left\ unexplained\ (errors).$$

Use: $2\sum_{i=1}^{n}(y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) \quad = 2\sum_{i=1}^{n} e_i(\hat{y}_i - \bar{y}) = 2(\sum_{i=1}^{n} e_i \hat{x}_i)\,\hat{B} - 2\bar{y}(\sum_{i=1}^{n} e_i) = 0$

- *This just shows that these equations show that the total variation in the data can always be split into the part explained by the regression model and the part left unexplained (errors).*

- R-Sqaured: Proportion of the total variation in y that is explained by the regression.

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

- Adjusted: same as R2 ut penalises adding extra predictors; more reliable for comparing models with different numbers of regressor.

$$\overline{R^2} = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

- The F-statistic: (OVERALL SIGNIFICANCE):

$$F = \frac{\frac{RSS_0 - RSS}{q}}{\frac{RSS}{(n-p-1)}} \text{ or } F = \frac{\frac{ESS}{p}}{\frac{RSS}{n-p-1}}$$

- q = number of restrictions (usually p).
  - ➔ Tests whether the model explains significantly more variation than just a constant term; a large F suggests the regression is jointly significant.

How RSS behaves as the number of predictors P increases?

Note to begin, remember in a regression we want to minimise the RSS as then our models predictors are as close as possible to the actual data y.

1. **Model with (P) predictors:** $y = X_p B + \varepsilon$
   - Its error measure is the Residual sum of squares:

$$RSS(p) = \left||y - X_p B|\right|^2 = \sum_{i=1}^{n} (y_i - x_i B)^2$$

2. **Now Model with (P+1) predictors:**
   - Add one more predictor z and define:
     i.e., it is the same as the model with p predictors plus a new predictor z.

$$X_{p+1} = [X_p \; z]$$

   - New model is hence:

$$y = X_{p+1}\theta + u$$

   - It's Residual sum of squares:

$$RSS(p+1) = \left||y - X_{p+1}\theta|\right|^2$$

3. **Restricted vs. unrestricted optimisation:**
   - ***Restricted model***: With P predictors:

$$\underbrace{min}_{B} \left||y - X_p B|\right|^2$$

     ➔ *i.e., we are minimising the RSS for the regression with p predictors.*
     ➔ Equivalent to restricting the $P + 1$ feature model so that the new coefficient z is zero!

$$\underbrace{min}_{\theta:\theta_{p+1}=0} \left||y - X_{p+1}\theta|\right|^2$$

   - ***Unrestricted model***: With P+1 predictors unrestricted and hence minimising:
     ➔ *i.e., here the coefficient z can be anything.*

$$\underbrace{min}_{\theta} \left||y - X_{p+1}\theta|\right|^2$$

   - **Key IDEA:** both cases, restricted (where P+1 z parameter is 0) and unrestricted (where P+1 z parameter can be anything) we are still just minimising the same thing

$$\sum (y_i - prediction)^2$$

4. **Key inequality:**
   - **If you allow yourself more freedom (more predictors, no restriction), you can always fit the data at least as well as before.**
   - Since the unrestricted optimisation gives you *at least as much flexibility* as the restricted one.

$$RSS(P+1) \leq RSS(P)$$

5. **Conclusion:**
   - **Training RSS** is weakly decreasing in (P)
     - ➔ **That is:**
       Adding another predictors can never increase training RSS; it either reduces it or leaves it unchanged (which is what we want as we want to minimise RSS).
       i.e., With more predictors, the model is more flexible → it can fit the data at least as well as before and hence RSS will at least satay the same if not decrease it, remember RSS is the error explained part.
   - However, this doesn't guarantee better test performance — **overfitting risk increases** as (P) grows.

Statistical Decision Theory: Loss Functions:

Introduction on Loss Functions:

When we build a model, we are really making decisions (predictions or classifications). To judge how good those decisions are, we use a loss function that measures how far off we are.

- **A loss function** is a mathematical tool used in machine learning to measure the discrepancy between a **model's predicted output** and **the actual (true) output**.

BEST MODEL: ONE THAT MINIMISES THE EXPECTED LOSS (average loss across the population)

Types of Loss Functions:

- **Suppose that we have a target variable Y to be predicted based on an input vector X using a model m.**
  - The prediction is:

$$\hat{Y} = \widehat{f_m}(X)$$

  - ➔ The **functional (math) form** of $\widehat{f_m}(X)$ depends on the model **m**.
  - ➔ The estimated parameters in $\widehat{f_m}(X)$ are estimated from the data.
- **Common Loss Functions and their solutions:**
  - **Define LOSS FUNCTION:** $L(Y, \hat{Y})$ to measure prediction error: *(Predict Y by $\widehat{f_m}(X)$)*

1. **Quadratic Loss**:

$$L = (Y - \hat{Y})^2 = (Y - f_m(X))^2$$

   - Penalizes large errors more
   - ➔ **Optimal predictor (estimator):** is the *conditional mean E(Y|X).*
     - This is what the OLS regression gives
2. **Absolute Loss**:

$$L = |Y - \hat{Y}| = |Y - f_m(X)|$$

   - Penalizes errors linearly
   - ➔ **Optimal predictor** (estimator): is the **conditional median** (minimised by Least Absolute Deviations, LAD).
3. **0-1 Loss (in classification) :**

$$L = 1(Y \neq \hat{Y})$$

   - ➔ **Optimal predictor (estimator):** is the **conditional mode** (most likely class given (X)).
4. **Asymmetric Loss**:

$$L_t = \begin{cases} \tau(Y - \hat{Y}), Y \geq \hat{Y} \ (under\ predicted) \\ \tau(\hat{Y} - Y), Y \leq \hat{Y} \ (over\ predicted) \end{cases}$$

- Penalizes over- and under-predictions differently.

➔ **Optimal predictor** is the **conditional quantile** at level (τ)

➔ **For Example: Key idea: the loss penalizes over-predicting and under-predicting differently.**

- If $\boldsymbol{\tau = 0.5}$, the loss is symmetric (like absolute loss).
- If $\boldsymbol{\tau > 0.5}$, under-predictions are penalized more than over-predictions.
- If $\boldsymbol{\tau < 0.5}$, over-predictions are penalized more

5. **Log-likelihood loss**:

$$\boldsymbol{L = -log\hat{p}(Y|X)}$$

- Common in probabilistic models

➔ **Optimal predictor** is the conditional distribution, estimated via Maximum Likelihood Estimation (MLE).

Optimal Solution: Expected Loss Minimisation:

- **Overall goal is to** minimise the EXPECTED LOSS or RISK That is:
  - Choose a prediction function $\{f_m(X)\}$ that minimises the **expected loss** over the population

$$\boldsymbol{E(L) = E_X[L\ (Y, f_m(X))|X = x)]}$$

➔ $\boldsymbol{E[L\ (Y, f_m(X))|X = x)]}$**:** this is the average loss of X = x is fixed.

- *i.e., we are only considering the randomness in Y given this X = x.*
- In other words, given a specific x what is the expected loss of our prediction f(x).
  e.g., House w/ 3 bedrooms (X = 3), calculate average squared error of prediction.

➔ **The expectation** $\boldsymbol{E_X[*]}$**,** now hence averages over all possible values of X in the population.

- This gives the OVERALL expected loss (risk) for the model.
  e.g., Do this for all possible houses, weighted by how common each type is.

Example of optimal Solution for Quadratic Loss Case:

- **If the loss is Quadratic error:** $L(Y, f_m(X)) = (Y - f_m(X))^2$
  - **Let** $f_m(X) = g$
- **Then the expected loss is:** $\boldsymbol{E(L) = E[(Y - g)^2|X = x]}$
- **Finding the Optimal Prediction (FOC):**
  - Take the Derivative of the expected loss with respect to $f_m(X)$

$$FOC: \frac{\partial}{\partial f(X)} \boldsymbol{E}[(Y - g)^2|X = x]] = 0$$

  - Because expectation is linear, derivative passes through:

$$E_X\left[\frac{\partial}{\partial g}(Y - g)^2|X = x\right] = 0$$

  - Compute derivative inside and simply:

$$-2E_x[(Y - g)|X = x] = 0$$
$$E_X[(Y - g)|X = x] = 0$$
$$E_X[Y|X = x] - E_X[g|X = x] = 0$$
$$[E_X[g|X = x] = E_X[Y|X = x]$$

- (g) is a function of X, so conditional on X it is fixed, and its expectation is itself.

$$g = E_X[Y|X = x]$$

  - Since, $f_m(X) = g$ and $E_X[Y] = E[Y|X]$
    ➔ The optimal predictor is hence:

$$f_m(X) = E[Y|X]$$

- **Intuition:** The best predictor (the one that minimised expected loss) under squared error is always the expected value of Y given X.

Classification: Loss Function

- **In classification, we want to predict the class of an observation:**

$$\hat{Y} = G(X)$$

- where $G(X)$is the **decision rule** (what class we assign for each input X).
  - ➔ To know how "bad" a prediction is, we use a **loss function**!
  - ➔ For **C classes**, the loss is usually written as a **C × C loss matrix** L:

1. Each row = the **true (actual) class**
2. Each column = the **predicted class**
3. Each entry $L(c, g)$= **cost of predicting class** g **when the true class is** c

- ***For Example: 3 financial portfolio classes (Low, Medium, High)***
  - ➔ $L_{L\hat{M}}$ = cost of predicting **Medium** when it is actually **Low**.
  - ➔ $L_{L\hat{L}} = 0$ because predicting correctly has no cost

Commonly Used Loss Function: 0–1 Loss

- A simple loss function is **0–1 loss**:

$$L(Y, G(X)) = \begin{cases} 1, if\ predicted\ class\ \neq true\ class\ i.e., Y \neq \hat{Y} \\ 0, if\ predicted\ class = true\ class\ i.e., Y = \hat{Y} \end{cases}$$

- Interpretation:
  - ➔ Loss = 1 → wrong prediction (misclassification)
  - ➔ Loss = 0 → correct prediction
- **Example: for a 3*3 Matrix for above financial portfolio:**
  - ➔ $L = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

i.e., This is just a unit penalty for any misclassification.

Binary Choice: Confusion matrix:

- **Consider a binary classification problem with two categories: 1 (called positive) and 0 (called negative).**
- **A confusion matrix** counts the number of true negatives, false positives, false negatives, and true positives (ideally use test data to avoid overfit).

Actual	Classification (Prediction)		Total
	$\hat{Y} = 0$	$\hat{Y} = 1$	
$Y = 0$	True negatives (TN)	False positives (FP)	Actual N
$Y = 1$	False negatives (FN)	True positives (TP)	Actual P
Total	Predicted N	Predicted P	

Decision rule: other loss functions:

- **Under 0–1 loss, all errors are treated equally, a wrong prediction costs 1, and a correct one costs 0.**
  - **The Bayes Classifier:** For a binary problem ($Y \in \{0,1\}$), the decision rule is:

$$G(x) = \begin{cases} 1 \text{ if } P(Y=1 \mid X=x) \geq 0.5 \\ 0 \text{ if } P(Y=1 \mid X=x) < 0.5 \end{cases}$$

➔ This means we classify an observation as 1 when it's more likely than not to belong to that class.

➔ The 0.5 threshold is optimal only when false positives and false negatives have equal cost!

- **When Other loss functions are needed:**
  - In practice, some mistakes are more costly than others.
    - ➔ ***For example, in fraud detection:***
      - ***a) Predicting legitimate as fraud → investigation cost***
      - ***b) Predicting fraud as legitimate → large fraud loss***
  - When such costs differ, the 0.5 rule is no longer optimal — we adjust the loss function and use a different threshold (usually lower when false negatives are more costly).

Generalised decision rules:

- We introduce a **decision threshold** τ:

$$G_\tau(x) = \begin{cases} 1 \text{ if } P(Y=1 \mid X=x) \geq \tau \\ 0 \text{ if } P(Y=1 \mid X=x) < \tau. \end{cases}$$

  - The **Bayes classifier** is the special case where $\tau = 0.5$ (optimal under 0–1 loss but not in general !!)
    - ➔ The **choice of** τdepends on the **relative costs** of false positives and false negatives — lowering τ makes the model more sensitive (predicts 1 more often).

      i.e., a lower $\tau = when\ false\ negatives\ are\ more\ costly$.
      - Increases sensitivity (fewer false negatives)
      - Classifies more cases as positive (1).

      i.e., a higher $\tau = when\ false\ positives\ are\ more\ costly$.
      - Increases specificity (fewer false positives)
      - Classifies less cases as positive (1).

Type equation here.

Loss Matrix:

- **A loss matrix** specifies the loss for each combination of actual and predicted outcomes:

Interpretation:

- $L_{TN}, L_{TP} \leq 0$**: gains (correct predictions)**
  - ➔ L_{TN}correctly predicting a negative (true negative)
  - ➔ L_{TP}: correctly predicting a positive (true positive)
  - ➔ These typically represent a **gain** or **no loss**, so we often set them to **0** (neutral) or even **negative** values to indicate a *benefit* from being correct.
- $L_{FN}, L_{FP} > 0$**: losses (incorrect predictions)**
  - ➔ L_{FN}: false negative — failing to detect a true positive
  - ➔ L_{FP}: false positive — incorrectly predicting a positive
  - ➔ These represent **costs** or **penalties**, so they are assigned **positive values**, indicating a *loss* when such errors occur.

Example (Fraud Detection):

- $L_{TP} = 0$: correctly flagging fraud → no penalty.

- $L_{TN} = 0$: correctly accepting a legitimate transaction.
- $L_{FP} = 1$: falsely flagging a legitimate transaction → investigation cost.
- $L_{FN} = 10$: missing actual fraud → large financial loss.

Sensitivity and Specificity: (basically seeing how well your classifier is working)

- **Sensitivity (True Positive Rate):** Fraction of actual positives correctly identified.
  - **True positives/ Actual positives**

$$\text{TPR} = \frac{TP}{TP + FN} = P(\text{Predicted } 1 \mid Y = 1)$$

- **Specificity (True Negative Rate):** Fraction of actual negatives correctly identified.
  - **True negatives/ actual negatives**

$$\text{TNR} = \frac{TN}{TN + FP} = P(\text{Predicted } 0 \mid Y = 0)$$

- **Trade-off:**
  - Lowering τ→ classify more cases as 1 → **sensitivity ↑, specificity ↓**
  - Raising τ→ classify fewer cases as 1 → **specificity ↑, sensitivity ↓**

Imbalanced Cases:

- In some situations, we care more about sensitivity than specificity and vice versa
- In cases like fraud detection or rare disease detection:
  - One class (***e.g., negatives***) dominates → **imbalanced data.**
  - Accuracy or specificity alone can be misleading (a trivial "always negative" classifier may appear highly accurate).
  - **Focus on sensitivity**, especially if missing positives is costly.
  - To reduce false negatives when $\boldsymbol{L_{FN} \gg L_{FP}}$**, we lower the optimal threshold** $\boldsymbol{\tau}$**,** making the classifier more likely to predict positives.
- In short, the threshold τ allows us to balance **sensitivity vs specificity** according to the **relative cost of errors and class imbalance**.

Back to Regression Problem: In Sample v Out of Sample Loss:

- Consider a regression model: $\boldsymbol{Y = f(X) + \varepsilon}$
- where:
  - $\boldsymbol{f(X)}$is the true underlying function,
  - $\boldsymbol{\varepsilon}$ is random noise with: $\boldsymbol{E(\varepsilon \mid X) = 0, E(\varepsilon^2 \mid X) = \sigma^2}$
  - $\boldsymbol{f}$is an estimator of the true function based on training data.
- We measure performance using quadratic loss in this example: $\boldsymbol{L(Y, \hat{f}(X)) = (Y - \hat{f}(X))^2}$

Out of Sample Loss:

- Let's consider **a new observation** $\boldsymbol{x_o}$ not used to fit $\boldsymbol{f}$: (o = out)

$$\boldsymbol{Y_o = f(x_o) + \varepsilon_o}$$

  - Here, ε_o is independent of the fitted estimator $\hat{f}$.
  - The **expected out-of-sample loss conditional** on $\boldsymbol{x_o}$is:

$$\boldsymbol{E_o[(Y_o - \hat{f}(x_o))^2 \mid X = x_o] = E_{Y_o|X=x_o}[(f(x_o) + \varepsilon_o - \hat{f}(x_o))^2]}$$

  - Expand the square:

$$E_{Y_o|X=x_o} = (\hat{f}(x_o) - f(x_o))^2 + E[\varepsilon_o^2] = (\hat{f}(x_o) - f(x_o))^2 + \sigma^2$$

- **Interpretation:**
  - The **expected out-of-sample loss is the sum of**:
    - ➔ **Squared estimation error**: how far $\hat{f}(x_o)$is from the true $f(x_o)$
    - ➔ **Irreducible noise:** σ^2, which cannot be predicted

IN sample Loss:

- Now consider **training data** x_{in}used to fit f**:** (in = in)

$$Y_{in} = f(x_{in}) + \varepsilon_{in}$$

  - Here, $\hat{f}(x_{in})$**is fitted using** Y_{in}, which contains the noise ε_{in}.
    - ➔ Therefore, $\hat{f}(x_{in})$**and** ε_{in}***are not independent that is are correlated.***
- The **expected in-sample loss conditional** on $x_{in}$is:

$$E_{in}[(Y_{in} - \hat{f}(x_{in}))^2 \mid X = x_{in}]$$

  - Expanding:

$$E_{Y_{in}|X=x_{in}}[(f(x_{in}) - \hat{f}(x_{in}) + \varepsilon_{in})^2] =$$

$$E_{Y_{in}|X=x_{in}}[(\hat{f}(x_{in}) - f(x_{in}))^2] + \sigma^2 - \mathbf{2\,Cov}(\hat{f}(x_{in}), \varepsilon_{in}|X = x_{in})$$

  - **Key point:**
    - ➔ The covariance term arises because $\hat{f}(x_{in})$is fitted on noisy data, and it is positively correlated with the noise ε_{in}.
    - ➔ This reduces the in-sample loss compared to the true expected loss — an effect known as **optimism or overfitting.** **(i.e. highlight part above is the optimism).**

Summary Intuition:

	Out-of-Sample	**In-Sample**
Noise independence	Yes	No
Expected loss	$(\hat{f} - f)^2 + \sigma^2$	$(\hat{f} - f)^2 + \sigma^2 - 2Cov(\hat{f}, \varepsilon)$
Bias	None	Optimistically low (overfitting)!

Interpretation:

- **Out-of-sample loss** gives a realistic measure of prediction error on new data.
- **In-sample loss** underestimates true error due to overfitting — it is "optimistic" because the model has partially fit the random noise in the training data.

Bias and Variance Decomposition:

- **Sampling uncertainty:** Given an input x, the output $Y$is random because it depends on noise or randomness in data.
  - **We are interested in the expected squared error of our model prediction** $\hat{f}(x)$**:** (basically what is shown in both the out of sample and in sample)

$$E(f(x) - \hat{f}(x))^2$$

Step 1: Add and subtract the mean prediction

- We add and subtract $E[\hat{f}(x)]$inside the square (this is a standard trick to decompose error):

$$\boldsymbol{E\left(f(x)-\hat{f}(x)\right)^2 = E\left(f(x)-E\hat{f}(x)+E\hat{f}(x)-\hat{f}(x)\right)^2}$$

Step 2: Expand the square

Now expand $(A+B)^2 = A^2 + 2AB + B^2$, where,

- $A = f(x) - E\hat{f}(x)$(a constant with respect to the training data),
- $B = E\hat{f}(x) - \hat{f}(x)$(a random variable — depends on data).

$$\boldsymbol{E\left(f(x)-\hat{f}(x)\right)^2} = E\left[\left(f(x)-E\hat{f}(x)\right)^2 + 2\left(f(x)-E\hat{f}(x)\right)\left(E\hat{f}(x)-\hat{f}(x)\right) + \left(E\hat{f}(x)-\hat{f}(x)\right)^2\right]$$

Step 3: Take expectations

$$\boldsymbol{E\left(f(x)-\hat{f}(x)\right)^2} = E\left(f(x)-E\hat{f}(x)\right)^2 + 2E\left(f(x)-E\hat{f}(x)\right)\left(E\hat{f}(x)-\hat{f}(x)\right) + E\left(E\hat{f}(x)-\hat{f}(x)\right)^2$$

- When we take the expectation over the randomness in the training data, note that $E[\hat{f}(x)]$is a **constant** (the mean of model predictions).
  - The cross term (middle term) is zero since $E[E\hat{f}(x) - \hat{f}(x)] = 0$.
    - ➔ $2E\left(f(x)-E\hat{f}(x)\right)\left(E\hat{f}(x)-\hat{f}(x)\right) = 0$
  - So, we are left with:

$$\boldsymbol{E\left(f(x)-\hat{f}(x)\right)^2} = E\left(f(x)-E\hat{f}(x)\right)^2 + 0 + E\left(E\hat{f}(x)-\hat{f}(x)\right)^2$$

Step 5: Interpret

$$\boldsymbol{\left(f(x)-E[\hat{f}(x)]\right)^2 = \text{Bias}^2}$$

- how far the average prediction is from the true function.
  - (Error due to systematic assumptions of the model.)

$$\boldsymbol{E\left(\left(E[\hat{f}(x)]-\hat{f}(x)\right)^2\right) = \text{Variance}}$$

- how much the model's predictions vary around their mean when trained on different datasets.
  - (Error due to sensitivity to data.)

Final decomposition:

$$\boldsymbol{E\left(f(x)-\hat{f}(x)\right)^2 = Bias^2 + Variance}$$

- **Interpretation:**
  - Low bias → model captures the true pattern well.
  - Low variance → model predictions are stable across datasets.
  - High bias → underfitting.
  - High variance → overfitting.

Training Vs Test Data:

- The light blue curves show the training errors and red curves test loss.
  - Blue is training loss.

➔ As number of parameters increases then the loss decreases as the level of optimism increases. (the covariance terms we between the estimator and error will increase and hence constant decrease in the training loss).

- Red is test loss.
  ➔ The optimal is rather can see is made and will find a point where we reach the best number of parameters.

How to deal with this:

- In the ideal case of rich data, we can divide the data into two sets: training set and test set
  - **Training set:** A portion of your data used to **fit/estimate the model** (learn the parameters).
  - **Test set:** A different portion used to **measure the model's prediction error** (how well it generalizes).
- **Goal:** Pick the model with the **lowest prediction error** on the test set (not the training set).
- **Challenges**
  - In practice, **data is limited**, so splitting into train and test reduces how much data is used for training.
  - We can do better with:
    ➔ **Cross-validation:** repeatedly splitting data into train/test folds and averaging the performance.
    ➔ **Complexity-penalizing criteria:** like AIC, BIC, or regularization penalties, which let you use all data but still control overfitting.

Model selection criteria

Mallows Cp Criterion:

Idea:

- **The training loss** (*e.g., residual sum of squares in regression*) **tends to be optimistically small** compared to how well the model will perform on new data.
  - This difference is called **"optimism"**.
    ➔ This is because the model is "optimistic" — it fits the noise in the training data a bit. ***i.e., training error RSS underestimates the true test error because the model is fit to the data.***
- **Mallows C_p** estimates this optimism and adjusts the training error to predict **expected test error**.

Deriving Mallows Cp:

Step 1: Express optimism as covariance: (per above from in sample and out of sample loss section)

- **Optimism** measures how much training error underestimates test error:

$$\text{Optimism} = \mathbb{E}[\text{Test Loss} - \text{Train Loss}] = \left(\hat{f} - f\right)^2 + \sigma^2 - \left(\left(\hat{f} - f\right)^2 + \sigma^2 - 2Cov\left(\hat{f}, \varepsilon_{in}\right)\right)$$

$$= 2\sum_i \text{Cov}(\hat{f}(x_i), \epsilon_{in})$$

  ➔ $\hat{f}(x_i)$= predicted value for point i
  ➔ ϵ_{in}= true noise in Y_{in} (in sample)
  - This covariance captures how fitting to the training data reduces error artificially.

Step 2: Look at Linear regression:

- Given : $\boldsymbol{Y = X\beta + \epsilon}$ the least squares fit is:
  - $\boldsymbol{\widehat{Y} = X\widehat{B} = X(X'X)^{-1}X'Y = HY}$
    - ➔ **Where** $\widehat{B}$=(X'X)−1X'Y
    - ➔ **Where** : $H = X(X'X)^{-1}X'$is the hat matrix, mapping observed Y to the fitted values $\hat{Y}$.
- **Covariance with in-sample errors (between fitted and errors):**
  - $\boldsymbol{\widehat{Y} = H(XB + \epsilon) = HXB + H\epsilon = XB + H\epsilon}$
    - ➔ (because HX=X(X'X)−1X'X=XI=X)
    - ➔ The deterministic part is $X\beta$
    - ➔ The **random part** is $H\epsilon$.
  - *Looking at how the model's fitted values ($\hat{Y}$) "move with" the true random errors (ϵ):*

$$\boldsymbol{\text{Cov}(\widehat{Y}, \epsilon) = \text{Cov}(H\epsilon, \epsilon) = H\text{Cov}(\epsilon, \epsilon) = H\sigma^2}$$

    - ➔ *Since $H$is a constant matrix and Cov$(\epsilon) = \sigma^2 I$*
    - ➔ *So, the covariance matrix between $\hat{Y}$and $\epsilon$is simply $H\sigma^2$*
- **Diagonal elements (pointwise):**

$$\boldsymbol{\text{Cov}(\widehat{Y}_i, \epsilon_i) = \sigma^2 H_{ii}}$$

  - Diagonal element H_{ii} gives the covariance between fitted value $\boldsymbol{\widehat{Y}_i}$and its own noise term $\boldsymbol{\epsilon_i}$:
    - ➔ **Meaning:**
      - If $H_{ii}$is large, that point is far from the average of the data and has **high leverage**, meaning it strongly affects its own prediction, so its fitted value $\hat{Y}_i$is more tied to its own noise ϵ_i, which can make it **influential** + potentially **distort the regression line**.
- **Sum over all points:**

$$\boldsymbol{\sum_{i=1}^{n} \text{Cov}(\widehat{Y}_i, \epsilon_i) = \sigma^2\, \text{tr}(H) = \sigma^2 p}$$

  - $\text{tr}(\text{H}) = \text{p}$= number of estimated parameters (including intercept)
    - ➔ (The trace (sum of diagonal elements) of H always equal to the number of estimated parameters p)
  - Intuition: Each fitted value "inherits" randomness from ϵ proportional to leverage H_{ii}. Adding these across all data points gives total covariance = $\sigma^2 p$.
    - ➔ That total appears in the **Mallows** $C_p$correction: it quantifies how much the model's training error is *optimistically biased* downward because of overfitting to noise

Step 3: Expected test error:

- **Expected test RSS: From Step 1 remember we see 2 times COV and hence:**

$$\boldsymbol{\mathbb{E}[\text{Test RSS}] = \mathbb{E}[\text{Train RSS}] + 2\sigma^2 p}$$

  - **Intuition:** Add $2\sigma^2 p$ to the training RSS to account for model **complexity and optimism.**
    - ➔ **Where p = number of parameters:**

Step 4: Define Mallows Cp:

$$\boldsymbol{C_p = \frac{RSS}{\sigma^2} - n + 2p}$$

- σ^2 is estimated from the full (largest) model. (to provide reliable unbiased estimate)
- **Interpretation:**

- A good model has $C_p \approx p$
  (meaning its models complexity matches it predictive performance)
- If $C_p < p$, the model may be underfitting; if $C_p > p$, it may be overfitting.
- Smaller C_p generally indicates a better trade-off between fit and complexity, but ideally C_p should be close to p.

Key Takeaways

1. Training error alone is optimistic — it underestimates true error.
2. Mallows C_p corrects for optimism using model complexity (p).
3. Goal: Choose a model with small C_p close to p.

AIC:

- We want to choose a model that best approximates the true data-generating process but also avoids overfitting.

$$AIC = -2 * loglikelihood\ (\widehat{B_{MLE}}) + 2p$$

  - **First term**: how well the model fits the data (training loss).
  - **Second term (2p)**: penalty for model complexity (more parameters p).
    ➔ **p** = number of parameters in β (number of predictors).
- **Rule AIC**: choose the model with the **smallest AIC**.
  - Proposed by Hirotugu Akaike (1973)

AIC: Theoretical justification: (OPTIONAL/ DW CHAR LEAVE OUT)

- **Consider Data**: **(y, X)** and a **model m** with parameter **vector β of size p.**
- **Under model m, the Model-based density is p(y|β)** (this is the **likelihood function**).
- **The True (but UNKNOWN) data generating density is g(y)**
- **Goal:** pick model whose distribution **p(y|β)** is as close as possible to the true distribution **g(y).**
  - **The discrepancy between g(y)** (the truth) **and p(y|β)** (model) is **SMALL.**
- **Measure of closeness: Kullback–Leibler (KL) divergence**
  - KL divergence measures the discrepancy between two probability distributions:
    ➔ Consider for two probabilities distributions their densities g(x) and p(x):
    (2 ways to write it)

$$KL(g||p) = \int g(x) \log \frac{g(x)}{p(x)} dx \text{ OR } E_g[\log \frac{g(x)}{p(x)}] = E_g[log g(x) - log p(x)]$$

    ➔ **Note:** KL divergence is **not symmetric**: EL(g||P) ≠KL(p||g)
  - **Example (don't really need this in depth but anyways):**
    ➔ Two normal distributions: $g \sim N(u_1, \sigma_1^2)\ vs\ p \sim N(u_2, \sigma_2^2)$
    - Writing out the densities:

$$g(x) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\ [-\frac{(x-u_1)^2}{2\sigma_1^2}, p(x) = \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\ [-\frac{(x-u_2)^2}{2\sigma_2^2}$$

    - Taking the log inside KL:

$$\log g(x) = -\frac{1}{2}\log(2\pi\sigma_1^2) - \frac{(x-u_1)^2}{2\sigma_1^2}, \log p(x) = -\frac{1}{2}\log(2\pi\sigma_2^2) - \frac{(x-u_2)^2}{2\sigma_2^2}$$

    - Hence:

$$\log g(x) - \log p(x) = -\frac{1}{2}\log(2\pi\sigma_1^2) - \frac{(x-u_1)^2}{2\sigma_1^2} + \frac{1}{2}\log(2\pi\sigma_2^2) + \frac{(x-u_2)^2}{2\sigma_2^2}$$

$$\log g(x) - \log p(x) = \frac{1}{2}\log\left(\frac{\sigma_2^2}{\sigma_1^2}\right) - \frac{(x-u_1)^2}{2\sigma_1^2} + \frac{(x-u_2)^2}{2\sigma_2^2}$$

- Take expectation:

$$KL(g||p) = E_g\left[\frac{1}{2}\log\left(\frac{\sigma_2^2}{\sigma_1^2}\right) - \frac{(x-u_1)^2}{2\sigma_1^2} + \frac{(x-u_2)^2}{2\sigma_2^2}\right]$$

- Using Rules of expectations (not included here):

$$KL(g||p) = \frac{1}{2}\log\left(\frac{\sigma_2^2}{\sigma_1^2}\right) + \frac{\sigma_1^2 + (u_1-u_2)^2}{2\sigma_2^2} - \frac{\sigma_1^2}{2\sigma_1^2}$$

- This is the final closed form for KL divergence between two normal:

$$KL(g||p) = \log\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{\sigma_1^2 + (u_1-u_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

- **Define the models Loss:**
  - Define loss as KL divergence from true distribution g(y) to model-based distribution p(y|B):

$$\boldsymbol{L_m(B) = KL(g(y)||p(y|B)) = E_Y\left[\log\left(\frac{g(y)}{p(y|B)}\right)\right]}$$

  - ➔ This is the **distance** between the true distribution and the model's distribution.
  - ➔ We want a model m and parameters β that **minimize Lm(β) for all B**

- **Practical Problem:**
  - g(y) is unknown, so we can't compute KL divergence exactly.
    - ➔ **But** we can estimate β by **maximum likelihood (MLE)**:
    - ➔ Then we pick the model m that minimizes $\boldsymbol{E_{Y|X}[L_m(\hat{B}_{MLE})]}$
      - Where $\hat{B}_{MLE}$ is the MLE estimator of B (under model m)
- **Akaike's result:**
  - Akaike showed for large n

$$\boldsymbol{E_Y[L_m(\hat{B}_{MLE})] = \frac{1}{2}AIC + E_Y\{\log(g(y))\}}$$

  - Where:

$$\boldsymbol{AIC = -2 * loglikelihood\ p(y|\widehat{B_{MLE}}) + 2p = -2 * loglikelihood\ (\widehat{B_{MLE}}) + 2p}$$

  - ➔ As The second term $\boldsymbol{E_Y\{\log(g(y))\}}$ is constant across models (it doesn't depend on m)
  - ➔ Minimising AIC is equivalent to minimizing the expected KL loss.

Takeaway:

- **AIC ≈ (lack of fit) + (penalty for complexity).**
  - *Choosing the model with the smallest AIC balances fit and simplicity and asymptotically picks the model closest (in KL divergence) to the true data-generating process.*

BIC:

$$\boldsymbol{BIC = -2 * loglikelihood\ (\widehat{B_{MLE}}) + p(log\ n)}$$

- **First term:** how well the model fits the data (same as AIC).
- **Second term:** penalty for complexity; grows with log n (heavier penalty for large samples).

➔ **p:** number of parameters including the intercept.
➔ **n:** sample size.

- **Rule BIC:** choose the model with the **smallest BIC**.
  - Proposed by **Gideon Schwarz (1978)**.

BIC: Theoretical justification:

- **Consider a Model m with parameters β (size p), data (y,X)**
- **Posterior probability of model m**:

$$p(m|y) \propto p(m)p(y|m) = p(m)\int p(y|B,m)p(B|m)dB$$

1. $p(m|y)$: **Posterior probability**: updated probability of model m after seeing the data, which combines the prior and the likelihood.
2. $p(m)$: **Prior probability:** your belief in model m before seeing the data
3. $p(y|m)$: **marginal likelihood** the probability of observing the data assuming this model.
4. $p(y|B,m)$: **likelihood**: probability of the data for a given parameter β in model m.
5. $p(B|m)$: **prior for parameters**: what we believe about β before seeing data.
6. $\int p(y|B,m)p(B|m)dB$: averages the likelihood over all possible parameter values (β) to get the overall probability of the data under the model.

BIC Approximation:

- In Bayesian statistics, all information about m is in its **posterior** $p(m|y)$
  - We want the model with the highest posterior probability.
- If we **assume a uniform prior on m (p(m) is a constant), then comparing the models is just comparing p(y|m) which per above is just the whole integral section:**
  - Using **Laplace approximation, it can be shown** for the integral when n is large gives:

$$-logp(m|y) \approx -logp\left(y|\widehat{B}_{MLE},m\right) + \frac{p}{2}logn = \frac{BIC}{2}$$

➔ Multiplying by 2 yields the familiar BIC:

$$BIC = -2 * logp\ (y|\widehat{B_{MLE}},m) + (log\ n) * p$$

➔ First term = how well the model fits the data (log-likelihood at MLE).
➔ Second term = complexity penalty (grows with number of parameters p)

  - **KEY IDEA:**
    ➔ Thus, maximizing the posterior $p(m|y)$is (approximately) equivalent to **minimizing BIC**.
    ➔ **Interpretation:** Smaller BIC → better balance between model fit (likelihood) and complexity (number of parameters).

AIC or BIC? Maybe Hanna-Quinn

$$AIC = -2 * loglikelihood\ (\widehat{B_{MLE}}) + 2p$$
$$BIC = -2 * loglikelihood\ (\widehat{B_{MLE}}) + (log\ n) * p$$

- Both are used to **compare models**, balancing **fit** and **complexity**.
  - **BIC penalizes complexity more heavily** (because log n > 2 for moderate/large n).
    ➔ **Consistency:** BIC can identify the "true model" as n→∞ (if a true model exists).
  - **Small samples:** Practitioners often prefer **AIC** when n is small because BIC's heavy penalty may exclude useful variables

Hannan-Quinn (HQIC):

$$HQIC = -2 * loglikelihood\ (\widehat{B_{MLE}}) + 2p * log(log\ n)$$

- HQIC is a compromise between AIC and BIC, with a penalty slightly larger than AIC but smaller than BIC.

Takeaway:

- **AIC** → better for prediction, small n
- **BIC** → consistent, heavier penalty for complexity
- **HQIC** → middle ground

Summary:

- **Model & Method selection**: functional form, dataset, model averaging.
- **Loss → target**: square mean; absolute median; asymmetric quantile.
- **Classification**: 0–1 loss for unequal costs ⇒ pick threshold τ; balance sensitivity–specificity.
- **Generalisation**: training loss is optimistic and needs to be penalised.
- **Selection Criteria**: Mallow Cp, AIC, BIC, HQIC;