Business 4720 - Class 11

Supervised Machine Learning using Regression and Classification Models

Joerg Evermann

Faculty of Business Administration
Memorial University of Newfoundland
jevermann@mun.ca

This Class

What You Will Learn:

- Introduction to Statistical Learning
- Introduction to Regression Models
- Introduction to Classification Models

Based On

Gareth James, Daniel Witten, Trevor Hastie and Robert Tibshirani: *An Introduction to Statistical Learning with Applications in R*. 2nd edition, corrected printing, June 2023. (ISLR2)

`https://www.statlearning.com`

Chapters 2, 3, 4, 5

Trevor Hastie, Robert Tibshirani, and Jerome Friedman: *The Elements of Statistical Learning*. 2nd edition, 12th corrected printing, 2017. (ESL)

`https://hastie.su.domains/ElemStatLearn/`

Chapters 2, 3, 4, 7

Kevin P. Murphy: *Probabilistic Machine Learning – An Introduction*. MIT Press 2022.

`https://probml.github.io/pml-book/book1.html`

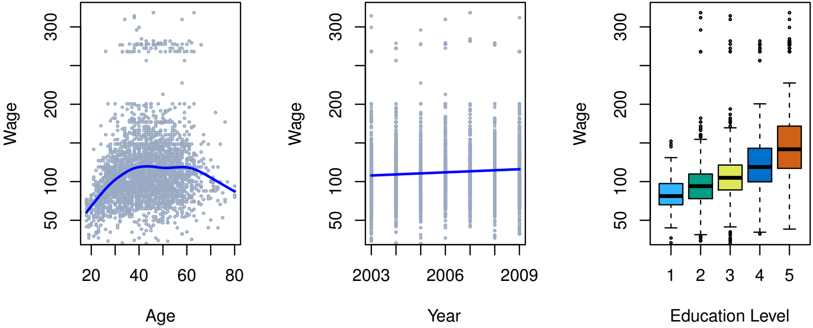
Chapters 4, 6, 9, 10, 11

Supervised Learning

- **Inputs** x ("predictors", "independent variables", "features") can predict **Output** y ("target", "response", "dependent variable")
  - May assume a functional relationship $y = f(x) + \epsilon$
- **Train** a statistical **model** using data where both inputs and outputs are known ("training data")
  - Approximate f by some function $\hat{f}$
  - "Fit" a model to data
- Parametric ("model-based") methods **learn** the **parameters** of a model for **optimal** prediction. They assume a functional form for $\hat{f}$
- Non-parametric methods do *not* assume a functional form and are more flexible
- **Predict** outputs of new observations using trained model $\hat{f}$

MEMORIAL UNIVERSITY

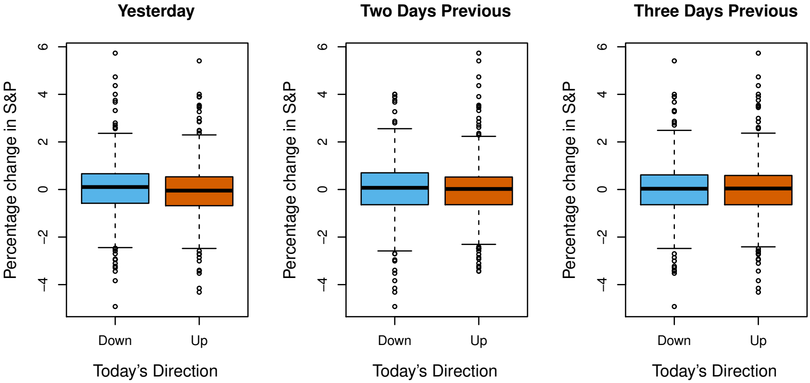
- Predicts **quantitative** output values
- Model quality measured by difference between actual and predicted



Source: ISLR2 Figure 1.1

Classification

- Predicts **categorical** or **qualitative** output values
- Model quality measured by proportion of mis-classification



Source: ISLR2 Figure 1.2

Regression Methods – Examples

Parametric Methods

- **Linear Regression**
- **Ridge and Lasso Regression**
- Principal components regression
- Non-linear regression
- Neural networks

Non-Parametric Methods

- **K-Nearest-Neighbours (KNN)**
- Regression trees
- Smoothing splines
- Multivariate adaptive regression splines
- Kernel regression

Classification Methods – Examples

- Decision trees
- Random forests
- Bayesian networks
- Support vector machines
- **Neural networks**
- **Logistic regression**
- Naive Bayes
- Probit model
- Genetic programming
- **K-Nearest-Neighbours (KNN)**

MEMORIAL UNIVERSITY

Prediction and Explanation

Explanation

- Identifying causal mechanisms
- Testing causal hypotheses or explanations
- *Inference* to *population parameters* (points, intervals)
- Form of relationship between inputs and outputs is important (parsimony, ease of interpretation)

Prediction

- Predict outputs for new observations
- Point or interval predictions, predictive distributions
- Focus on specific observations/cases
- Form of relationship between inputs and outputs is not important (may be complex, difficult to interpret)

MEMORIAL UNIVERSITY

Prediction and Explanation [cont'd]

Explanation	**Prediction**
Causation	Association
Theory	Data
Retrospective	Prospective
Bias	Variance

Based on: Shmueli, G. (2010). To Explain or To Predict?. Statistical Science, 25, 289-310.

Hands-On Exercises

For each of the following problems, decide if it is a prediction or inference/explanation problem:

1. How do real estate prices vary with location and age?
2. What is the most important predictor of real estate prices?
3. What is the expected sales price for a house at 310 Elizabeth Ave?
4. Is the month of the sale an important predictor of real estate prices?
5. Calculate the difference in expected sales prices for the house at 310 Elizabeth Avenue when sold in August and February
6. When should a house be sold to achieve the best price?

Model Quality — Regression

Optimization Objectives:

- MSE (mean squared error)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

(MSE is susceptible to outliers)

- MAE (mean absolute error)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{f}(x_i)|$$

MEMORIAL UNIVERSITY

Model Quality — Regression

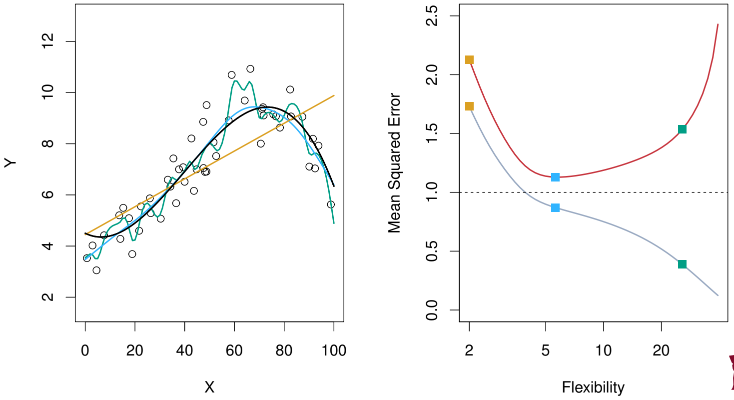
Evaluation focus is on unseen test data, not training data

- Train on past stock market info, but predict future stock performance
- Train on previous patient info, but predict future patient outcomes
- Train on past real estate prices, but predict future prices

*Separate **training data** from **test data** to evaluate model quality (”holdout sample”)*

Degrees of Freedom

- How much a function can be adapted to fit training data
- Number of independently ("freely") adjustable parameters



Source: ISLR2 Fig 2.9

Quality of Fit [cont'd]

Overfitting

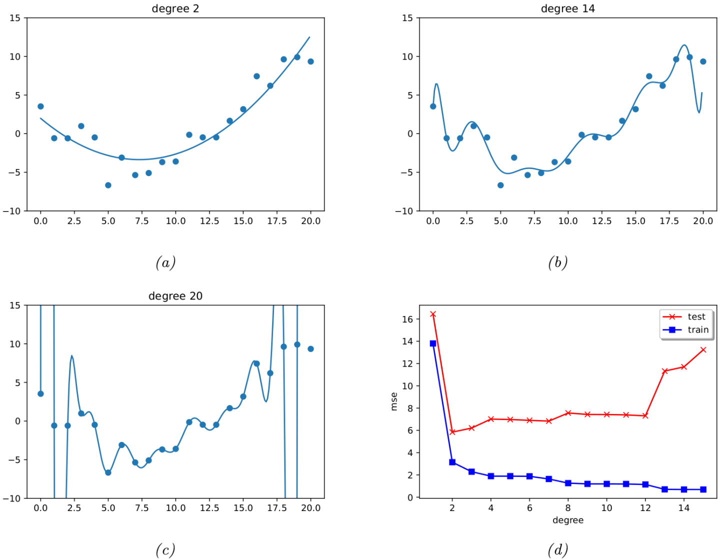
- Small training error
- Large testing error
- Model exploits random idiosyncrasies of the data set

Underfitting

- Large training error
- Large testing error
- Model is insufficiently able to fit true pattern in data (too simple, inflexible)

MEMORIAL UNIVERSITY

Overfitting with Polynomial Expansions



Source: Murphy Figure 1.7

Bias and Variance

Recall: Expected Value

$$E[X] = \sum_{i=1}^{\infty} x_i p_i \qquad \text{discrete random variable}$$

$$E[X] = \int_{-\infty}^{\infty} x f(x) dx \qquad \text{continuous random variable}$$

(for uniform distributions or unweighted observations $p_i = p_j \; \forall i, j$ so that $E[X] = \frac{1}{n} \sum_{i=1}^{\infty} x_i$, i.e. expectation = mean)

Recall: Variance

$$Var[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

(for zero-centered variables $E[X] = 0$ so that $Var[X] = E[X^2]$)

Bias and Variance Decomposition

Example using mean squared error loss

$$
\begin{aligned}
MSE &= E[(y - \hat{f})^2] && \text{(unweighted)} \\
&= E[y^2 - 2y\hat{f} + \hat{f}^2] \\
&= E[y^2] - 2E[y\hat{f}] + E[\hat{f}^2]
\end{aligned}
$$

$$
\begin{aligned}
E[\hat{f}^2] &= E[\hat{f}^2] - E[\hat{f}]^2 + E[\hat{f}]^2 \\
&= Var[\hat{f}] + E[\hat{f}]^2
\end{aligned}
$$

$$
\begin{aligned}
E[y^2] &= E[(f + \epsilon)^2] \\
&= E[f^2] + 2E[f\epsilon] + E[\epsilon^2] \\
&= f^2 + 2f \cdot 0 + \sigma^2 && (f \text{ is not random and } E[\epsilon] = 0) \\
&= f^2 + \sigma^2
\end{aligned}
$$

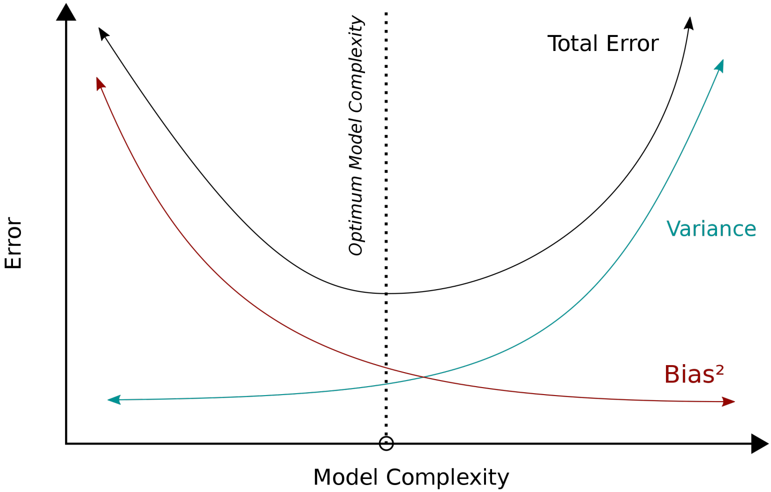
Bias and Variance Decomposition [cont'd]

Example using mean squared error loss

$$
\begin{aligned}
E[y\hat{f}] &= E[(f+\epsilon)\hat{f}] \\
&= E[f\hat{f}] + E[\epsilon\hat{f}] \\
&= E[f\hat{f}] + E[\epsilon]E[\hat{f}] \\
&= E[f\hat{f}] + 0 \cdot E[\hat{f}] \\
&= fE[\hat{f}]
\end{aligned}
$$

$$
\begin{aligned}
MSE &= f^2 + \sigma^2 - 2fE[\hat{f}] + Var[\hat{f}] + E[\hat{f}]^2 \\
&= (f - E[\hat{f}])^2 + \sigma^2 + Var[\hat{f}] \\
&= Bias[\hat{f}]^2 + \sigma^2 + Var[\hat{f}]
\end{aligned}
$$

Bias and Variance Trade-Off



https://commons.wikimedia.org/wiki/File:
Bias_and_variance_contributing_to_total_error.svg

Bias and Variance Trade-Off

Bias

- Model (assumptions) error
- *Bias*$[\hat{f}]$ is the error introduced by a wrong/simplified model
- **High bias**: Model is too simple to represent true relationship $\rightarrow$ **Underfitting**

Variance

- Training data error due to model complexity
- *Var*$[\hat{f}]$ is the variability between training data sets (samples)
- **High variance**: Model is too complex and exploits random noise in training data $\rightarrow$ **Overfitting**

Bias and Variance Trade-Off [cont'd]

Irreducible Error

- Unmeasured variables
- Measurement error
- σ^2 cannot be predicted from x_i so cannot be reduced.

Bias and Variance Trade-Off [cont'd]

- *Explanation* focuses on bias reduction (i.e. find the ”true” functional form)
- *Prediction* focuses on variance reduction (functional form is irrelevant).
- High variance models are complex, but complex models need not have high variance.
- High bias (simple models) does not imply large prediction error
- Lower prediction error does not imply low bias (simple models)

Model Quality — Classification

Error Rate

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

where $I(\cdot)$ is the *identity function* that is 1 if its argument is true, 0 otherwise.

- Training error rates
- Testing error rates

Bayes Classifier

Classifier

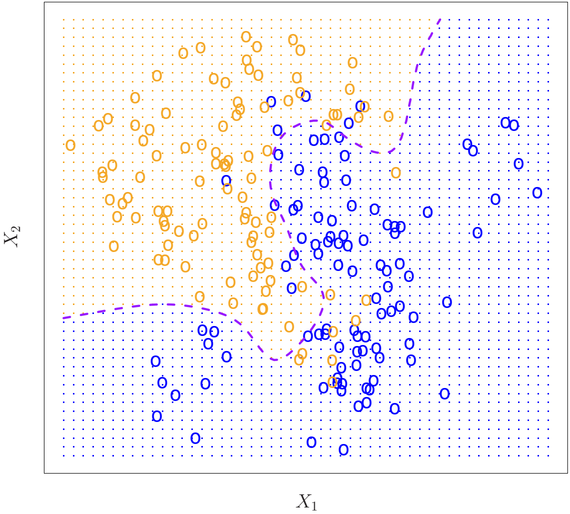
Assign each observation to the most likely class given its predictor values

$$\underset{j}{\operatorname{argmax}} \Pr(Y = j | X = x_0)$$

Error Rate

$$1 - E\left(\underset{j}{\operatorname{argmax}} \Pr(Y = j | X)\right)$$

Bayes Decision Boundary



Source: ISLR2 Figure 2.13

Bayes Classifier [cont'd]

- Bayes classifier is an *ideal* classifier
- Bayes error rate is lower bound, irreducible error
- Conditional probabilities are unknown in practice
- Estimation introduces error

Example — K-Nearest-Neighbour (KNN)

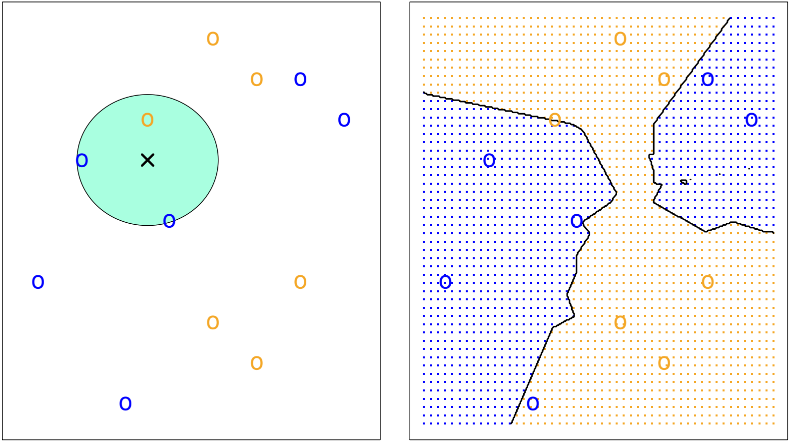
- Identify set of K points closest to observation x_0 called N_0

$$\Pr(Y = j | X = x_0) = \frac{1}{K} \sum_{i \in N_0} \mathrm{I}(y_i = j)$$

  where $\mathrm{I}(\cdot)$ is the identity function that is 1 if its argument is true, and 0 otherwise.

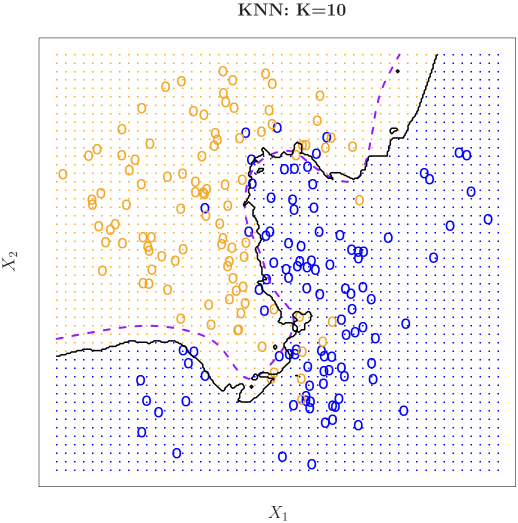
- Classify in class of highest probability

K=3



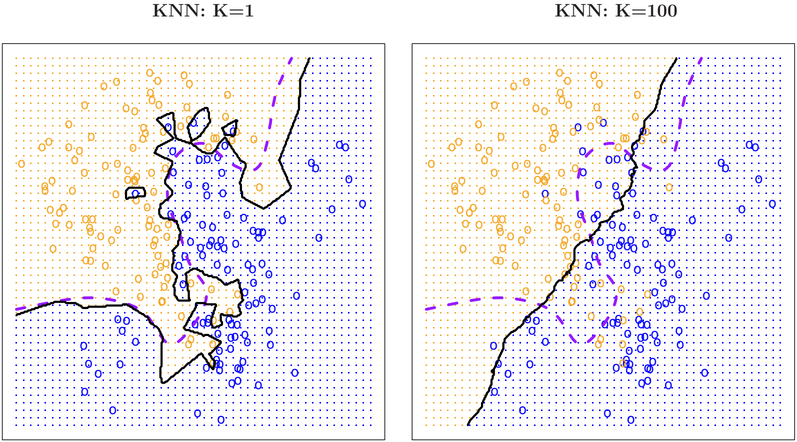
Source: ISLR2 Figure 2.14

KNN and Bayes Classifier



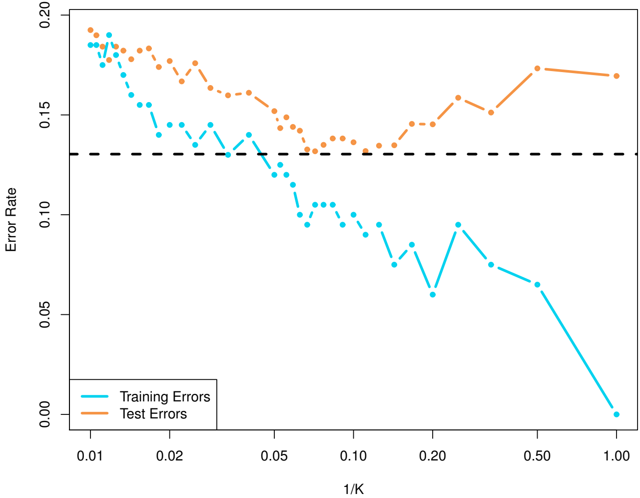
Source: ISLR Figure 2.15

KNN Quality



Source: ISLR2 Figure 2.16

KNN Error Rates



Source: ISLR2 Figure 2.17

Hands-On Exercise – KNN

The table below provides a training data set containing six observations, three predictors, and one qualitative response variable

Obs.	X_1	X_2	X_3	Y
1	0	3	0	Blue
2	2	0	0	Blue
3	0	1	3	Blue
4	0	1	2	Yellow
5	-1	0	1	Yellow
6	-1	1	1	Blue

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbours.

1. Compute the Euclidean distance (”L2-norm”) between each observation and the test point
2. What are your prediction with $K = 1$? With $K = 3$? Why?
3. If the Bayes decision boundary is highly non-linear, would you expect the best value for K to be large or small? Why?

Adapted from ISLR Exercise 2.7

Binary Classification Quality — Confusion Matrix

Decision rule: $\Pr(\text{default=Yes}|X = x) > 0.5$ (Bayes)

		True default status No	Yes	Total
Predicted	No	9644	*252*	9896
default status	Yes	*23*	81	104
	Total	9667	333	10000

Source: ISLR2 Table 4.4

- Overall error rate: 2.75%
- Of the defaulters, only 24.3% were correctly predicted ("**sensitivity**") (81/333), error rate 75.7%
- Of the non-defaulters, 99.8% were correctly predicted ("**specificity**"), error rate 0.02%

MEMORIAL UNIVERSITY

Confusion Matrix – Adjusting Thresholds [cont'd]

Decision rule: $\Pr(\text{default=Yes}|X = x) > 0.2$

		True default status		
		No	Yes	Total
Predicted	No	9432	138	9570
default status	Yes	235	195	430
	Total	9667	333	10000

Source: ISLR2 Table 4.5

- Overall error rate: 3.73%
- Sensitivity = 58.6%;
- Specificity = 97.6%

MEMORIAL UNIVERSITY

Confusion Matrix [cont'd]

		True class		
		No (-)	Yes (+)	Total
Predicted	No (-)	True Neg. (TN)	False Neg. (FN)	N^*
class	Yes (+)	False Pos. (FP)	True Pos. (TP)	P^*
	Total	N	P	

Binary Classification Model Quality

- Sensitivity, **Recall**, Hit Rate, True Positive Rate

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = 1 - FNR$$

- **Specificity**, Selectvitity, True Negative Rate

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR$$

- **Precision**, Positive Predictive Value

$$PPV = \frac{TP}{TP + FP} = 1 - FDR$$

- Negative Predictive Value

$$NPV = \frac{TN}{TN + FN} = 1 - FOR$$

Binary Classification Model Quality [cont'd]

- Miss Rate, False Negative Rate

$$FNR = \frac{FN}{P} = \frac{FN}{FN + TP} = 1 - TPR$$

- Fall-out, False Positive Rate

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} = 1 - TNR$$

- False Discovery Rate

$$FDR = \frac{FP}{FP + TP} = 1 - PPV$$

- False Omission Rate

$$FOR = \frac{FN}{FN + TN} = 1 - NPV$$

Binary Classification Model Quality [cont'd]

- **Accuracy** (= 1 - Error Rate)

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **F1 Score** (harmonic mean of precision and recall)

$$F1 = 2 \times \frac{PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$$

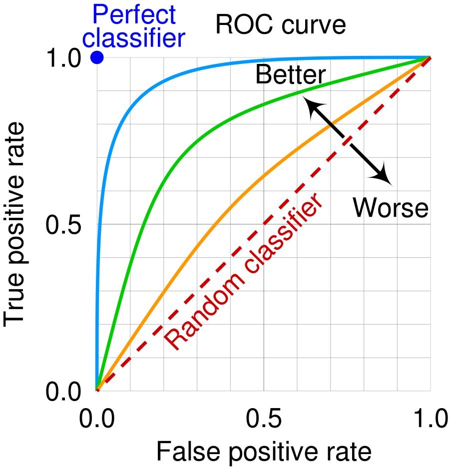
- False Discovery Rate

$$FDR = \frac{FP}{FP + TP} = 1 - PPV$$

- False Omission Rate

$$FOR = \frac{FN}{FN + TN} = 1 - NPV$$

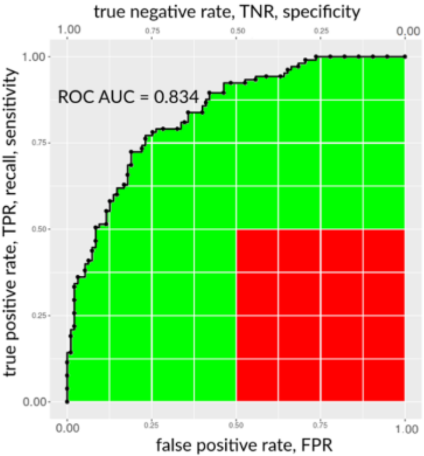
ROC: Receiver Operating Characteristic



https://commons.wikimedia.org/wiki/File:Roc_curve.svg

Binary Classification Model Quality [cont'd]

AUC: Area Under (ROC) Curve



https://commons.wikimedia.org/wiki/File:ROC_curve_example_highlighting_sub-area_with_low_sensitivity_and_low_specificity.png

Hands-On Exercise – Basic Calculations

1. Compute Precision and Recall for the two confusion matrixes above
2. Computer Accuracy and F1 values for the two confusion matrixes above
3. Plot the two points for this classifier in an ROC space/diagram. Are they above or below the diagonal?

MEMORIAL UNIVERSITY

Hands-On Exercise – Interpretation Challenge

Given the following results from a machine learning model:

- Precision: 0.75
- Recall: 0.60
- Accuracy: 0.80

Answer the following questions:

1. What percentage of identified positives are actually positive?
2. What percentage of actual positives are identified by the model?
3. What percentage of the total classifications were correct?

Hands-On Exercise – Adjusting Thresholds

Consider a binary classification task with the following confusion matrix at a certain threshold:

- TP: 150, FP: 50
- FN: 30, TN: 200

Discuss how adjusting the classification threshold might affect precision, recall, and accuracy. What happens if the threshold is increased or decreased?

Multi-Class Classification Model Quality

		True class			
		0	1	2	Prob
Predicted Class	0	4	2	0	$q_0 = 6/24 = .25$
	1	1	5	2	$q_1 = 8/24 = .33$
	2	2	0	8	$q_2 = 10/24 = .42$
	Prob	$p_0 = 7/24 = .29$	$p_1 = 7/24 = .29$	$p_2 = 10/24 = .42$	

- **Overall Accuracy**: sum(diag(.)) / sum(.) $= 17/24 = .71$

Multi-Class Classification Model Quality

Reduction to Binary Classification

- ”One vs. Rest” (OvR), ”One vs. All” (OvA), ”One against All” (OaA)
- Consider each class in turn as ”positive” class, consider all others as ”negative” class

MEMORIAL UNIVERSITY

Multi-Class Classification Model Quality [cont'd]

Micro-Averaging

- Count and sum TP, FP, FN over all classes
- Use the total TP, FP, FN to calculate Precision and Recall

- Gives equal weight to each instance
- May overemphasize performance of a majority class when it dominates the data set

For multi-class classification, micro-average precision equals micro-average recall and equals accurary

MEMORIAL UNIVERSITY

Multi-Class Classification Model Quality [cont'd]

Macro-Averaging

- Calculate precision and recall for each class (OvR)
- Average precision and recall, optionally weighting each class by its true count of instances

- Appropriate when all classes are equally important
- Appropriate for imbalanced data sets so all classes contribute
- May mask poor performance on important minority classes
- May lower overall performance due to low performance on small or unimportant classes

Hands-On Exercises

For the multi-class confusion matrix above,

1. Compute precision and recall for each class
2. Compute the macro-averages of precision and recall
3. Compute the micro-averages of precision and recall and show that they equal the accuracy

Multi-Class Classification Model Quality [cont'd]

- Dissimilarity between two probability distributions (information theoretic motivation)
  - True probability distribution over classes p_i
  - Predicted probability distribution over classes q_i
- **Cross-entropy**:

$$H(p, q) = -\sum_i p_i \log q_i$$

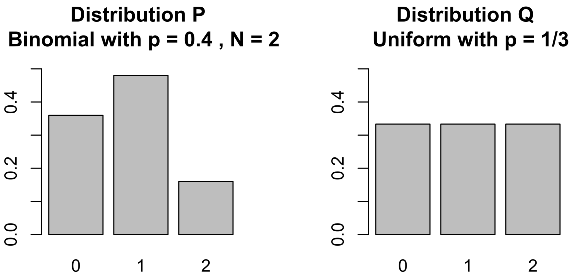
- **Kullback-Leibler (KL) divergence**:

$$\begin{aligned} D_{KL}(P||Q) &= \sum_i p_i \log\left(\frac{p_i}{q_i}\right) \\ &= \sum_i p_i \log p_i - \sum_i p_i \log q_i \\ &= -H(p, p) + H(p, q) \end{aligned}$$

MEMORIAL UNIVERSITY

Hands-On Exercises – Cross-Entropy & KL Divergence

https://commons.wikimedia.org/wiki/File:Kullback-Leibler_distributions_example_1.svg



Tip: Binomial distribution: $\Pr(P = k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$

1. Calculate the cross-entropy of P and Q
2. Calculate the entropy of P
3. Calculate the KL divergence of P and Q

Hands-On Exercises – Cross-Entropy

4. Calculate the cross-entropy and KL-divergence for the multi-class confusion matrix above
5. Given two probability distributions P and Q over a discrete set of events, where $P = [0.1, 0.4, 0.5]$ and $Q = [0.2, 0.3, 0.5]$, calculate the cross-entropy $H(P, Q)$ and the KL-divergence $D_{KL}(P||Q)$.

Hands-On Exercise – Cross-Entropy in Binary Classification

In a binary classification task, you have the following probability distributions for the actual labels (P) and predicted labels (Q):

- P = [1, 0] (the actual class is positive)
- Q = [0.7, 0.3] (the model predicts a 70% chance of being positive)

Calculate the cross-entropy loss for this scenario.

Hands-On Exercise – KL Divergence in Practice

Consider a scenario where you are comparing two models predicting weather conditions (sunny, cloudy, rainy). The actual distribution of weather conditions (P) and the predictions made by two models (Q1 and Q2) over a week are as follows:

- P = [0.5, 0.3, 0.2]
- Q1 = [0.4, 0.4, 0.2]
- Q2 = [0.6, 0.2, 0.2]

1. Calculate the KL divergence for both models relative to the actual distribution.
2. Which model is closer to the actual distribution based on the KL divergence?

MEMORIAL UNIVERSITY

Review Questions – Cross-Entroy & KL Divergence

1. Define cross-entropy and explain its significance in machine learning, especially in classification tasks.
2. Discuss how cross-entropy can be used to evaluate the performance of a classification model.
3. Define Kullback-Leibler divergence and explain its relationship with cross-entropy.
4. Discuss how KL divergence is used in machine learning models, especially in the context of model optimization and feature selection.

Resampling Methods

Goals

- Unbiased assessment of true classification error
- Generalization to unseen values

Model Selection

Estimate the predictive performance (error) of different models in order to choose the best one

Model Assessment

Having chosen a final model, estimate is prediction error on new data (generalizability)

Validation Set Approach (”Holdout” Method)

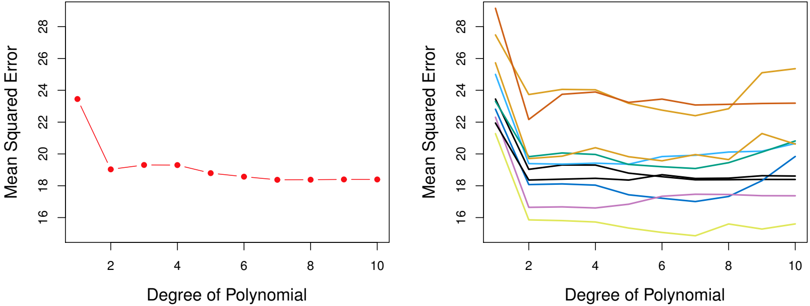
Procedure

- Randomly divide data:
  - **Training data**: Train each model
  - **Validation data**: Test each trained model
  - **Test data**: Evaluate the selected final model
- Typical split: 50% Training, 25% Validation, 25% Testing

Characteristics

- Validation error can be highly variable, depending on the split of data
- Validation error may overestimate actual error (bias), because of the smaller training set

Validation Set Approach (”Holdout” Method)



Source: ISLR2 Figure 5.2

Leave One Out Cross-Validation (LOOCV)

Procedure

1. Select one test observation
2. Train model with remaining $n - 1$ observations
3. Test the trained model on selected test observation
4. Repeat steps 1–3 n times with different test observations

$$CV = \frac{1}{n} \sum_{i=1}^{n} \mathrm{Err}_i$$

Characteristics

- Computationally expensive
- Stable results, no randomness
- Less overestimation (bias) of error rate

k-Fold Cross-Validation

Procedure

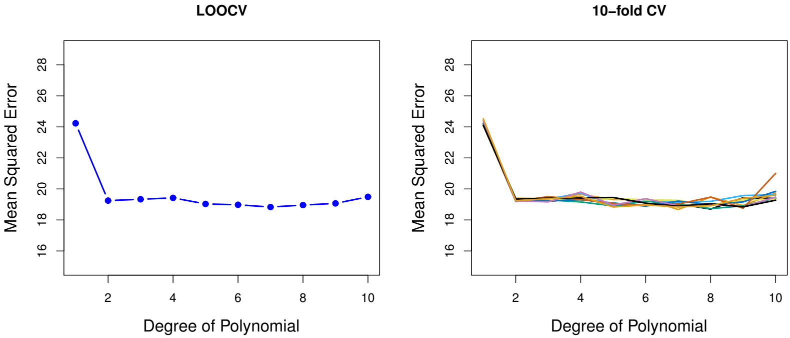
1. Randomly divide data into k sub-samples (”folds”)
2. Select one fold as test data
3. Train model on remaining $k-1$ folds
4. Test the trained model on test data fold
5. Repeat steps 2–4 k times using each fold as test data

$$CV = 1/k \sum_{i=1}^{k} \text{Err}_i$$

Characteristics

- Compromise between holdout method and LOOCV in terms of stability and computational expense
- Higher bias but lower variance of error estimate than LOOCV but lower variance than LOOCV
- **Typical** $k = 5$ **to** $k = 10$

k-Fold Cross-Validation



Source: ISLR2 Figure 5.4

Cross-Validation

To prevent ”information leakage” from training to test or validation data:

Important

- Initial analysis and predictor/feature selection must be done for each training set
- Data pre-processing (centering, scaling, outlier removal, etc.) must be done on each training set

MEMORIAL UNIVERSITY