THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Business Analytics (MM 5425)

L5. Model Evaluation

Dr. Yue (Katherine) FENG

Recap: Overfitting Problem!

Name	Balance	Age	Default
Mike	123,000	50	No
Mary	51,100	40	Yes
Bill	68,000	55	No
Jim	74,000	46	Yes
Dave	23,000	44	No
Anne	100,000	50	Yes
…	…	…	…
…	…	…	…
Henry	61,100	35	???
Amy	68,000	52	???
Allen	22,000	21	???
Tom	123,000	60	???
Jane	100,000	47	???

Data universe

Data you have (a subset/sample from the universe of the data)

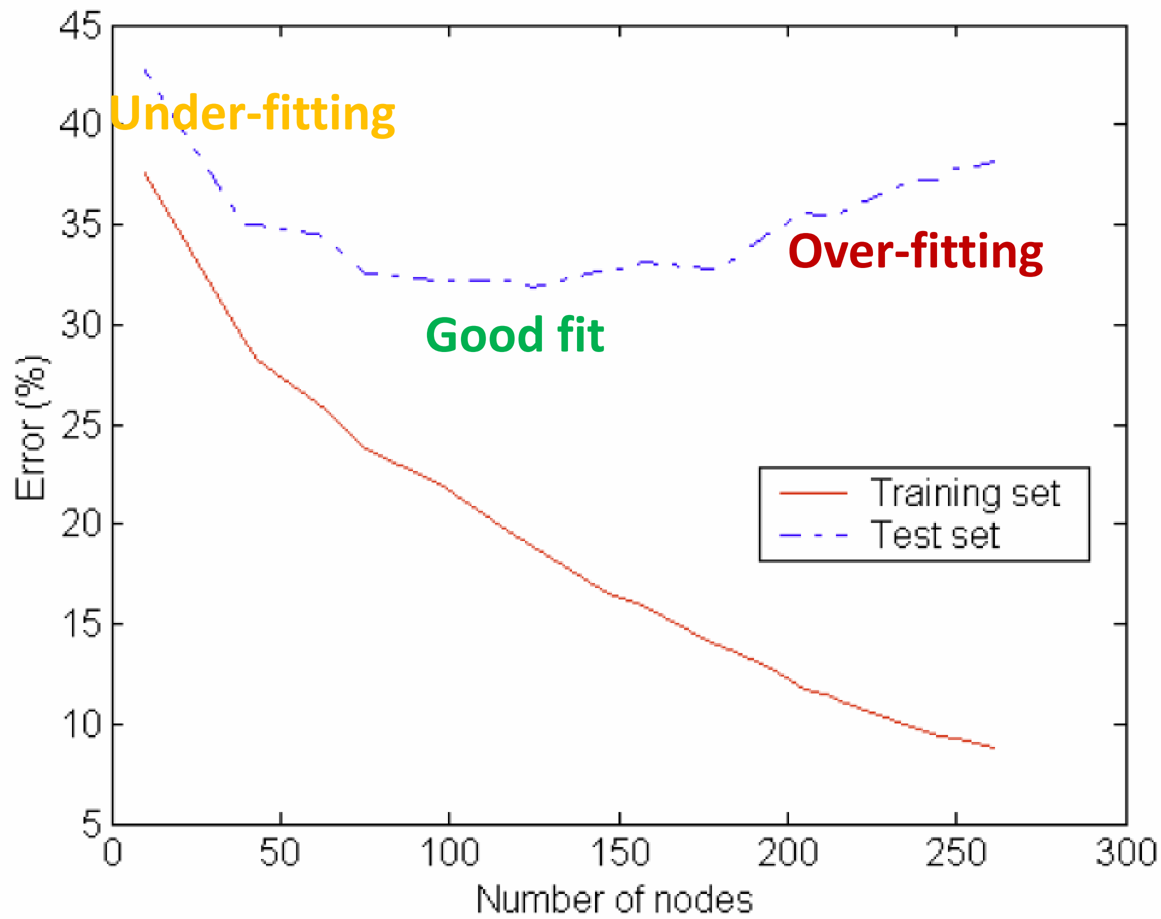
Build model

Model

Apply to new data

Overfitting: the pattern learned is too specific to be generalized to the universe.

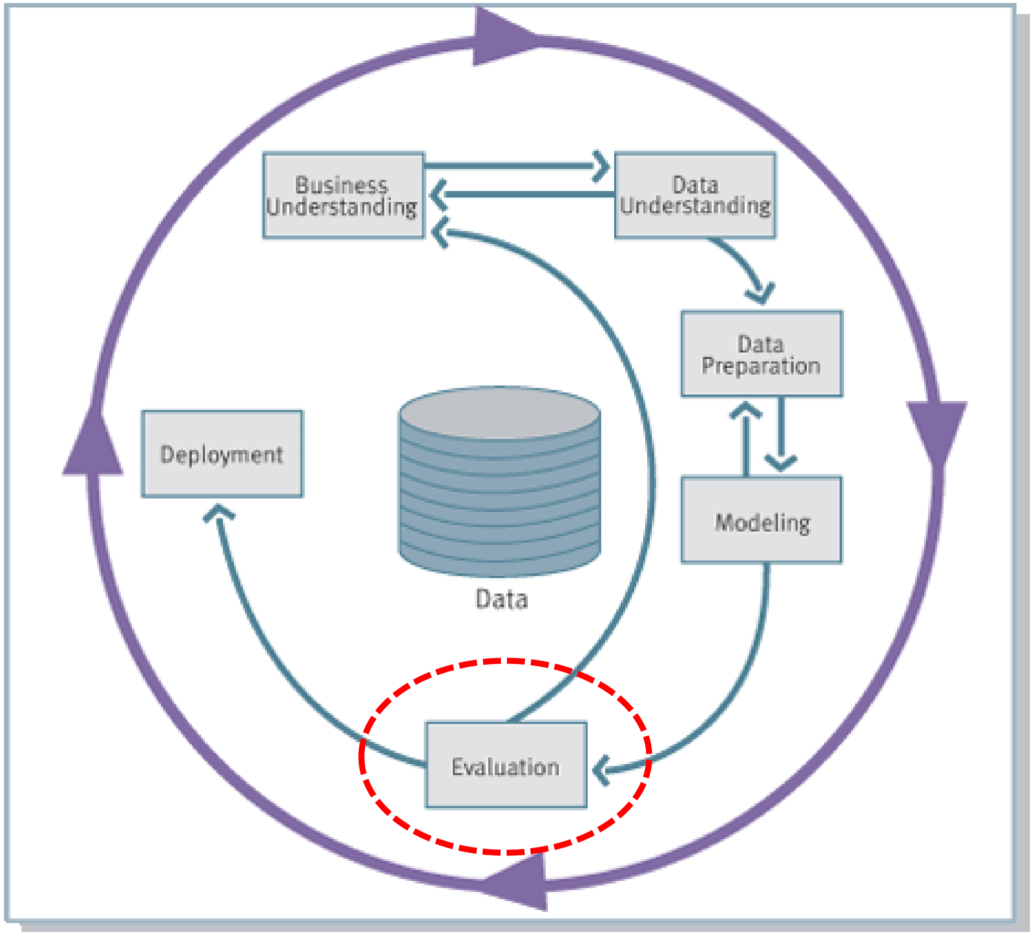
Recap: Symptom of Overfitting



Error rate = 1 - Accuracy

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Evaluation



Two Types of Evaluation

> Data-driven Evaluation

- e.g., training-testing split, cross-validation

> Domain Knowledge Evaluation

- using model as an interface between modelers and stakeholders
  - important to have a model that is comprehensible to stakeholders
  - compare with existing knowledge
  - get expert assessment of model

Agenda

I. From Holdout Evaluation to Cross-Validation

II. Accuracy

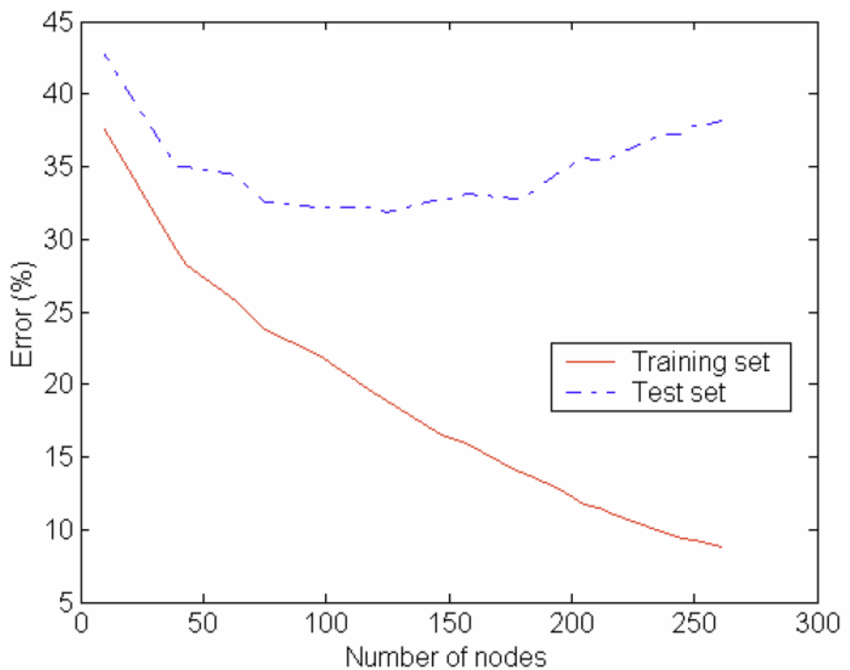
III. Confusion Matrix

IV. ROC Analysis

V. Performance Evaluation for Regression

From Holdout Evaluation to Cross-Validation

Holdout Evaluation

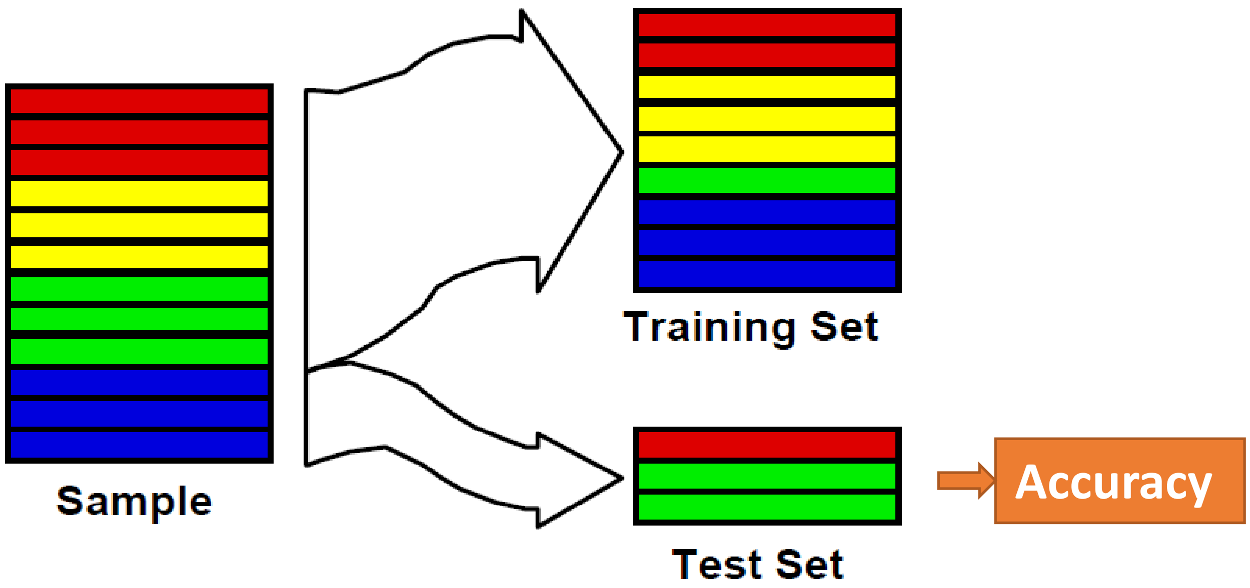


- Accuracy on training data: **"in-sample" accuracy**
- Accuracy on test data: **"out-of-sample" accuracy (generalization accuracy)**

> Given only one data set, we **hold out** some data for which we know the value of the target variable for evaluation.

> Holdout set for final evaluation is called the **test set**.

Simple Holdout Set



Do you trust the accuracy measure?

Questions:
1) what if by accident you selected a particularly easy/hard test set?
2) do you have an idea of the variation in model accuracy due to the split?

Drawbacks of the Simple Holdout Method

> The holdout estimate of the error can be highly variable across different splits.

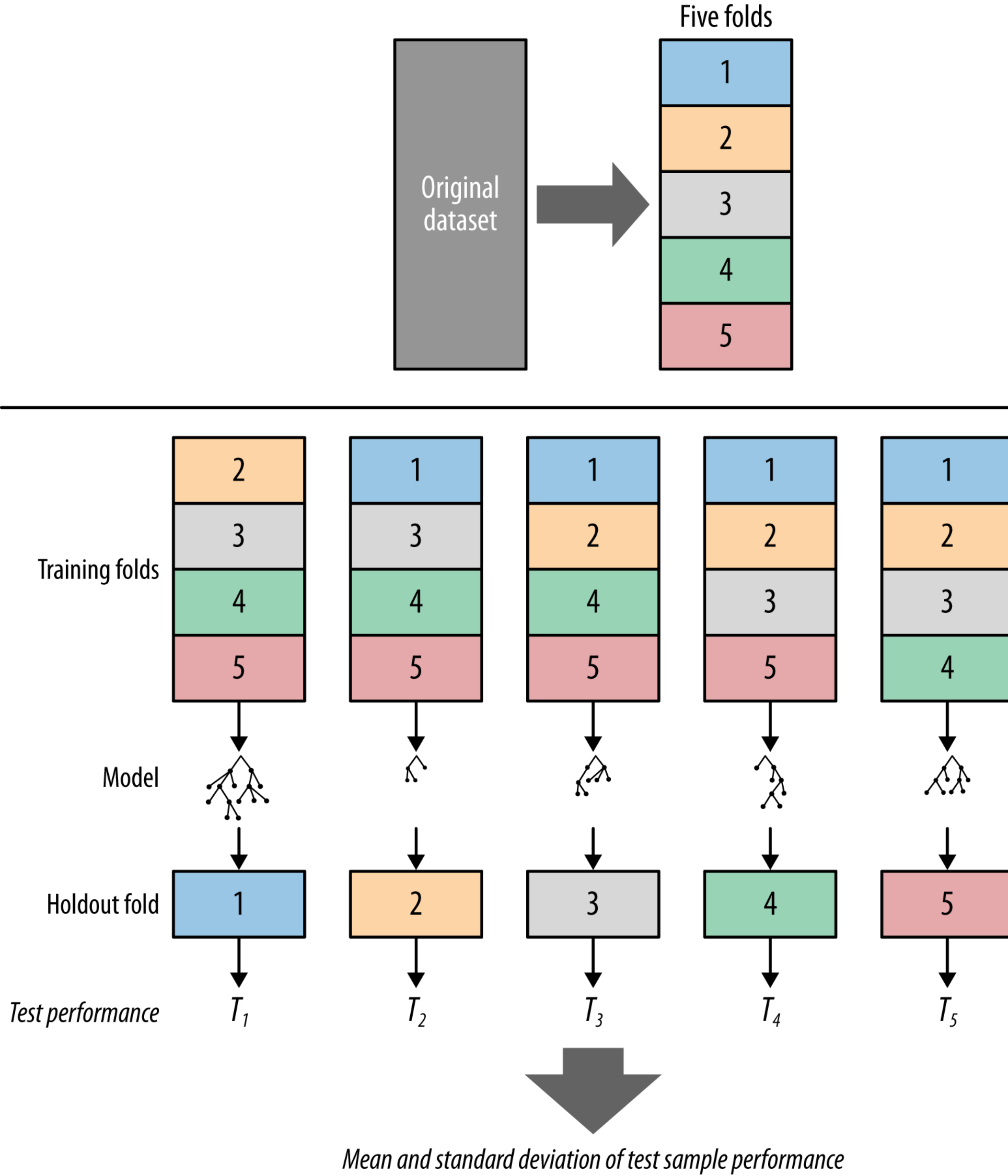
- Since it is a single train-and-test experiment, the holdout estimate of error rate will be misleading if we happen to get an “abnormal” split (e.g., very difficult test set).
- This is particularly true when we have a small dataset.

Solution: k-fold cross-validation!

Cross-Validation (CV)

> Better use of a limited dataset:
  - Partition data into ***k folds*** (randomly)
  - Run training/test evaluation ***k times***
  - For each of k experiments, use k-1 folds for training and a different fold for testing

> Estimate the true performance
  - Provide statistics on estimated performance (e.g., mean and variance)
  - Assess confidence in the performance estimate



- ✓ Each fold is tested once (rest are combined for training set)
- ✓ Tests on all data (each data point once)
- ✓ Can calculate average and variance of accuracy measure(s)

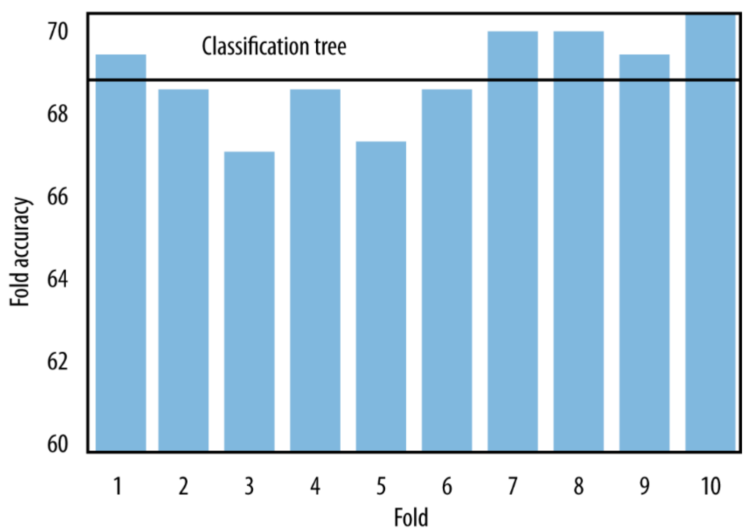
How Many Folds Are Needed?

> In practice, the choice of the number of folds depends on the size of the dataset
  - For large datasets, even 3-fold cross-validation will be quite accurate
  - For very sparse/small datasets, we may have to use more folds in order to train on as many examples as possible (e.g., k=N, leave-one-out cross-validation)

> A common practice for k-fold cross-validation is k=5 (by default in Python) or 10

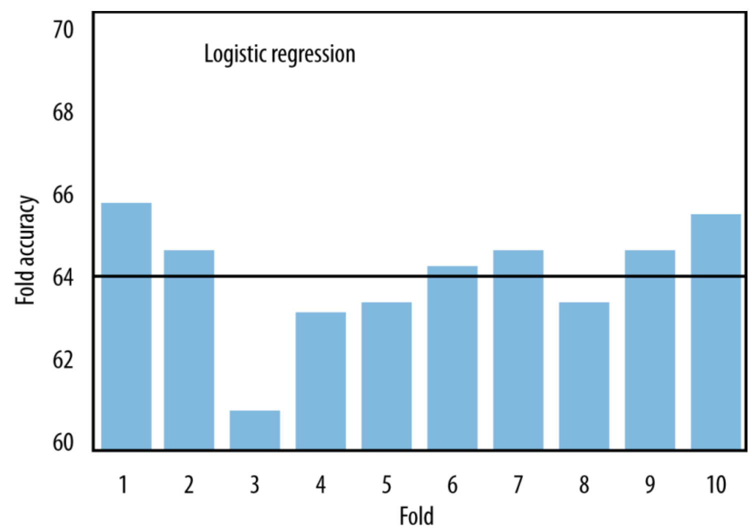
Statistics on Estimated Performance by Cross-validation

Accuracies of Classification Trees



Average accuracy: 68.6%
Standard deviation: 1.1

Accuracies of Logistic Regression



Average accuracy: 64.1%
Standard deviation: 1.3

This dataset contains 20,000 examples

Accuracy

Evaluation for Classification: Accuracy/Error rate

> **Accuracy:** the percentage of correct predictions

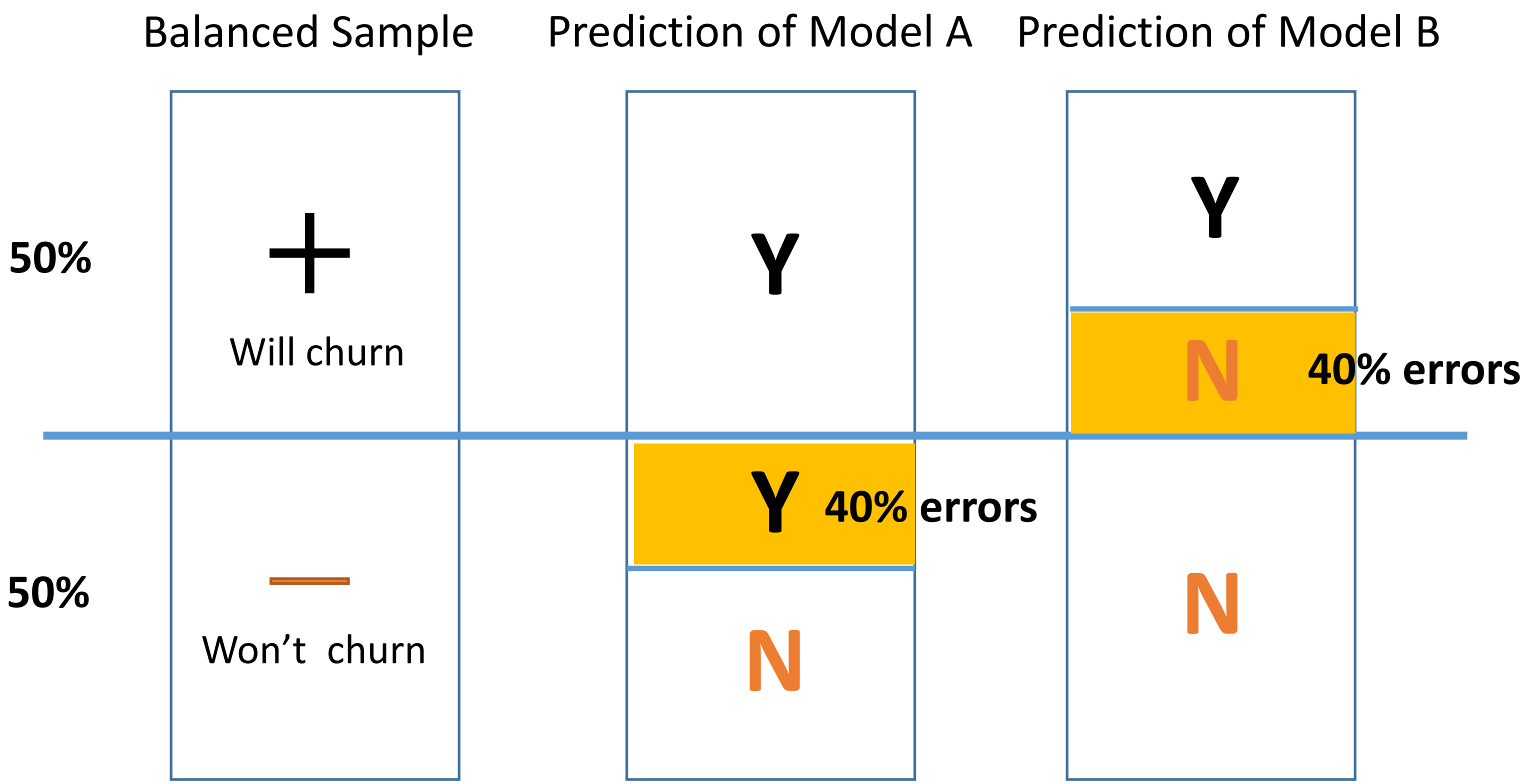
> **Error rate:** the percentage of incorrect predictions

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of instances in testing set}}$$

$$= 1 - \text{Error rate}$$

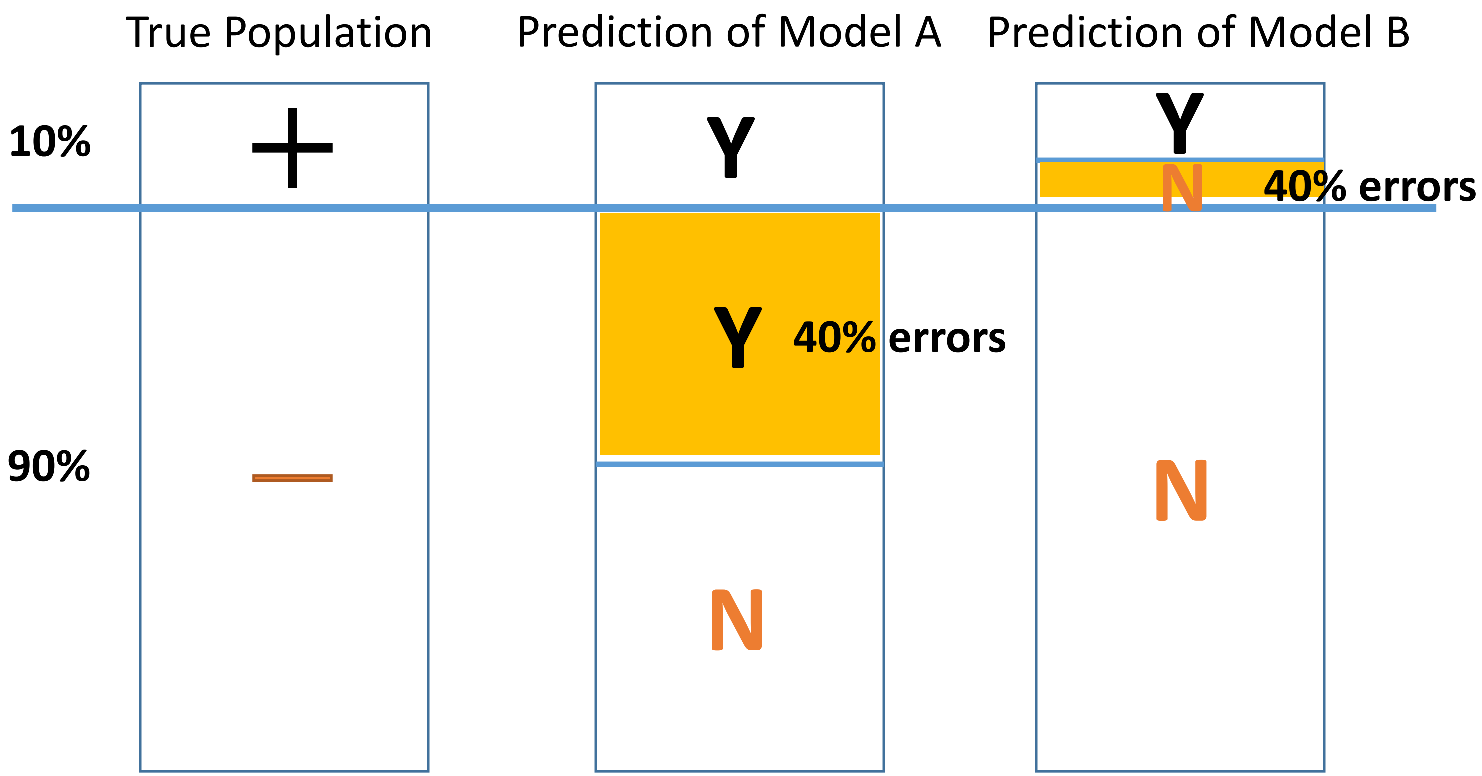
> **Too simplistic...and sometimes misleading (especially when the class distribution is unbalanced or skewed)**

Accuracy on a Balanced Sample



Both models correctly classify **80%** of the balanced population.

Accuracy on an Unbalanced Sample



Model A’s accuracy declines to **64%** while model B’s accuracy rises to **96%**.

Evaluation for Classification

> Naïve rule: majority-class classifier

- Classifying everyone as belonging to the majority class.
- Can be used as a **baseline** or **benchmark** for evaluating the performance of unbalanced samples.

Confusion Matrix

Confusion Matrix

> A table that is used to describe the performance of a classification model. Entries are **counts** of correct and incorrect predictions.

		Actual +	**Actual** -
Predicted	+	True + (TP)	False + (FP)
	-	False – (FN)	True – (TN)

Precision and Recall Measures

> **Precision** is the number of correctly classified positive examples divided by the total number of examples that are predicted as positive.

> **Recall** is the number of correctly classified positive examples divided by the total number of actual positive examples.

$$\text{Precision}(+) = \frac{TP}{TP + FP}$$

$$\text{Recall}(+) = \frac{TP}{TP + FN}$$

		Actual +	Actual -
Predicted	+	True + (TP)	False + (FP)
	-	False – (FN)	True – (TN)

Exercises

Decision Tree

Predicted \ Actual	+	-
+	8	20
-	2	970

Majority Class

Predicted \ Actual	+	-
+	0	0
-	10	990

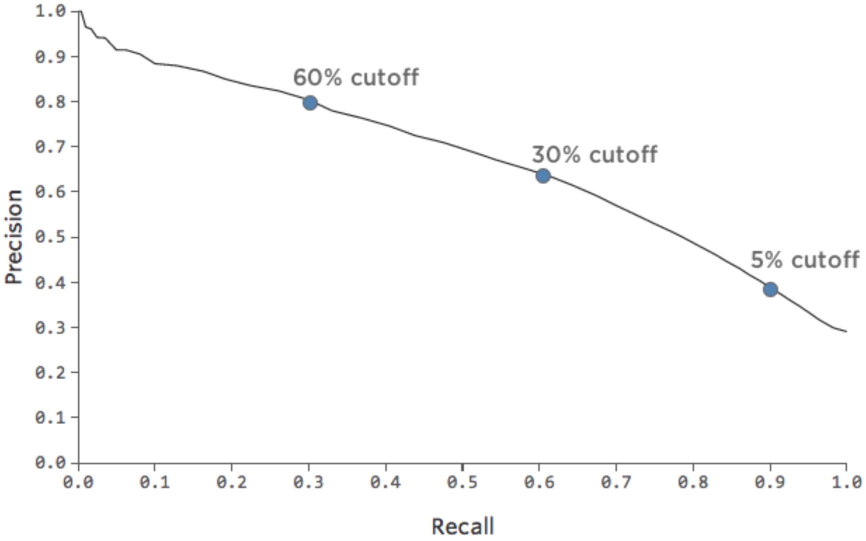
Q: What is the accuracy, precision, and recall for each model?

Other Evaluation Measure

> There is a tradeoff between precision and recall, where it is possible to increase one at the cost of reducing the other.

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

$$\text{F score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$



The higher the F1 score, the model performance is better.

Uneven Cost for Two Types of Errors

> In many applications, different types of errors have different costs.

- Wrongly approve a credit card application costs a lot more than wrongly deny an application.
- Wrongly filter out a good email costs a lot more than wrongly accept a spam.
- Wrongly diagnose an ill patient as normal costs a lot more than wrongly diagnose a normal people as ill.

		Actual +	Actual -
Predicted	+	True + (TP)	False + (FP)
	-	False – (FN)	True – (TN)

Bad Positives and Harmless Negatives

> Positive examples
  - An unusual condition is present: e.g., detect a disease, fraud case
  - Often rare and worthy of attention or alarm

> Negative examples
  - A normal or uninteresting outcome

> The number of mistakes on negative examples (false positive errors) may dominate, while the cost of mistakes on positive examples (false negative errors) will be **higher**.

		Actual +	Actual -
Predicted	+	True + (TP)	False + (FP)
	-	False – (FN)	True – (TN)

Using Expected Value to Frame Model Evaluation

Confusion Matrix (N=110)

	Actual +	Actual -
Predicted +	56	7
Predicted -	5	42

Cost/Benefit Matrix

	Actual +	Actual -
Predicted +	99	-1
Predicted -	0	0

EV=p(TP)*v(TP)+p(FP)*v(FP)+
p(TN)*v(TN)+p(FN)*v(FN)

ROC Analysis

ROC Analysis

> ROC - Receiver Operating Characteristic
- Developed in 1950s for signal detection theory to analyze noisy signals.
- A systematic way to evaluate the quality of probability estimates.
- Independent of class proportions and cost structures.

TPR and FPR

$$\text{True Positive Rate(TPR), Recall} = \frac{TP}{TP + FN}$$

$$\text{False Positive Rate(FPR)} = \frac{FP}{FP + TN}$$

		Actual	
		+	**-**
Predicted	**+**	**True + (TP)**	**False + (FP)**
	-	**False – (FN)**	**True – (TN)**

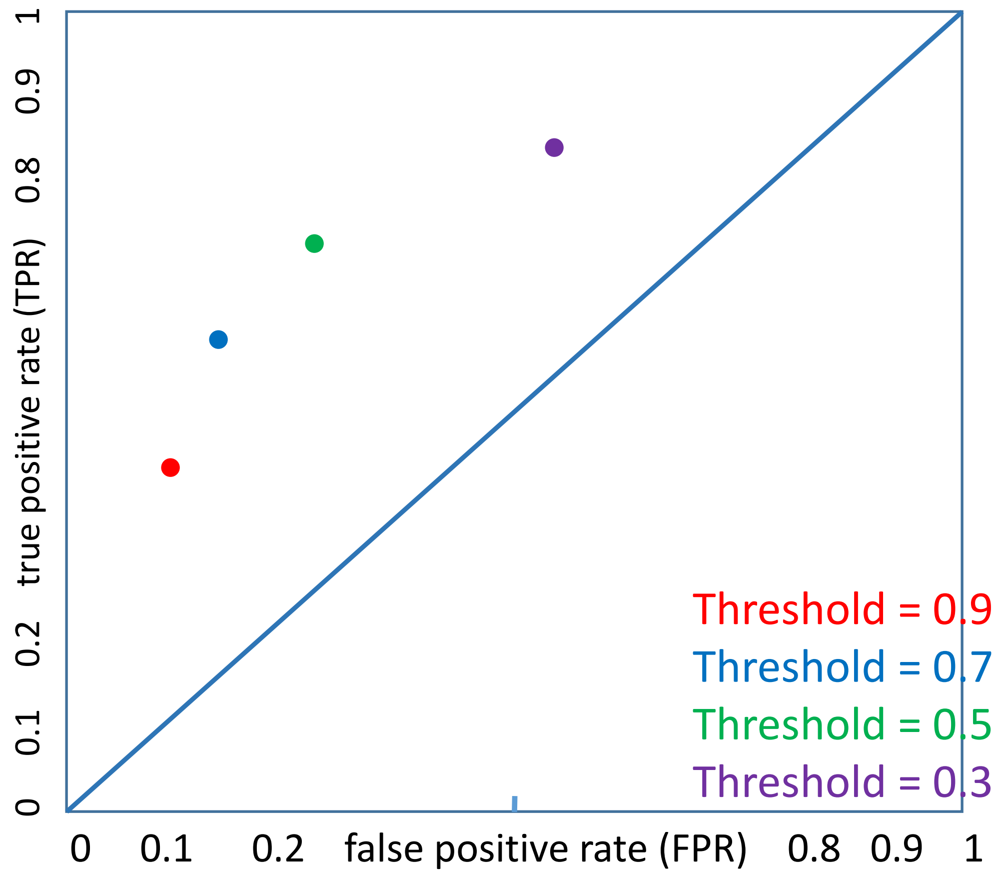
Decision Threshold

> Given the class probability estimates (PE), changing decision threshold could change TP or FP rate.

4 examples

PE for +	Prediction (0.5 threshold)	Prediction (0.8 threshold)	True value
0.3	-	-	-
0.55	+	-	+
0.75	+	-	-
0.9	+	+	-
True positive rate	1/1	0/1	
False positive rate	2/3	1/3	
Accuracy rate	2/4=0.5	2/4=0.5	

ROC Analysis

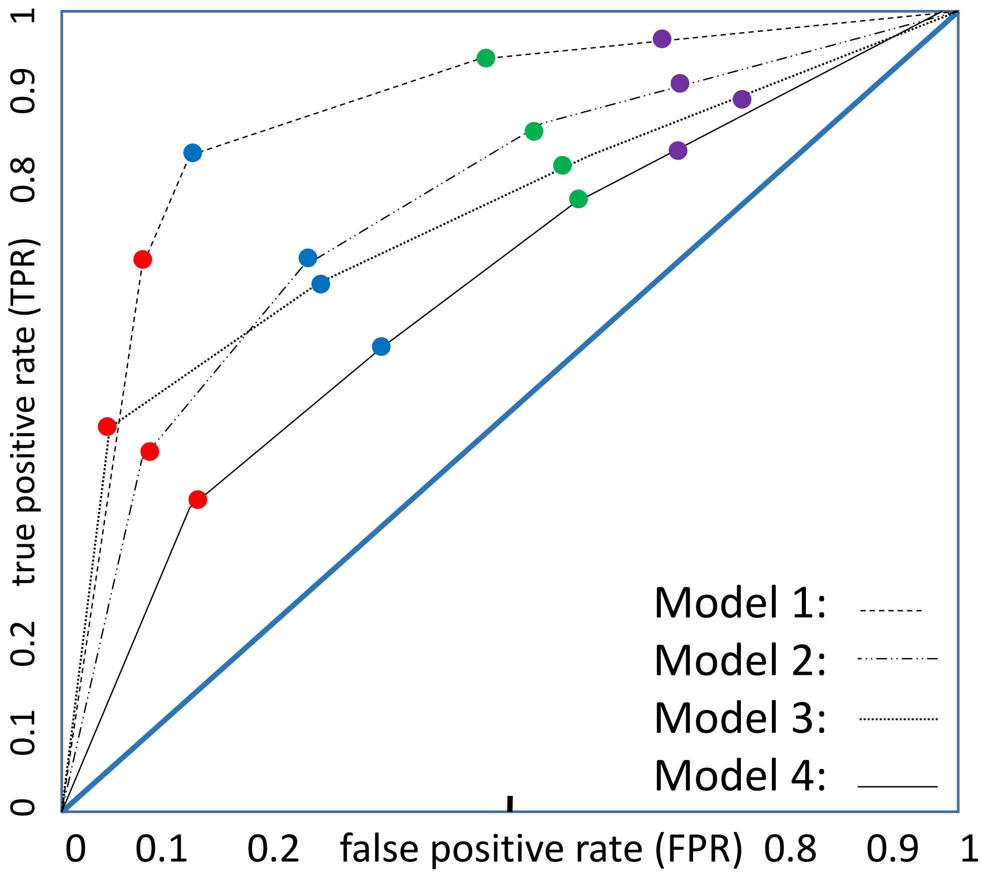


> Each decision threshold corresponds to a pair of TPR (on the y-axis) against FPR (on the x-axis), given a particular model.

> Changing the decision threshold changes the location of the point.

		Actual +	**Actual** -
Predicted	+	True + (TP)	False + (FP)
Predicted	-	False – (FN)	True – (TN)

ROC Curve



> Connecting dots to get a curve for a model.

> Different models have different curves.

> The **bigger** the area under ROC curve (**AuROC**), the **better** the model is.

What do the corners, (0,0), (0,1), and (1,1), and the diagonal line on ROC Curve mean?

AuROC vs. Accuracy

> Area under ROC curve: AuROC
  - Measure prediction performance under different decision threshold.

> AuROC is a **“deeper”** measurement
  - Measure the quality of probability estimates.
  - Better measure when uneven costs are unknown.
  - Better measure if the class proportion is unbalanced.

Performance Evaluation for Regression

Evaluation Measures for Regression

> Mean Squared Error (MSE): the average of squares of the differences between actual values and predicted values.

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

> Root Mean Squared Error (RMSE)

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

> Mean Absolute Error (MAE)

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

Evaluation Measures for Regression

> R squared (R^2): the proportion of the variation in the dependent variable (y) that is predictable from the independent variables (X).

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} = 1 - \frac{SS_{residual}}{SS_{total}}$$

- When the sum of squares of residuals $SS_{res} = 0$, $R^2 = 1$
- When all the predictions are $\bar{y}$, $R^2 = 0$ (baseline)
- Models with worse predictions than the baseline will have a negative R^2.

Confusion Matrix Output in Python

		Predicted	
		-(0)	+(1)
Actual	-(0)	True Negative	False Positive
	+(1)	False Negative	True Positive

Thank You !