DEPAUL UNIVERSITY

DSC478: Programming Machine Learning Applications

Roselyne Tchoua

rtchoua@depaul.edu

School of Computing, CDM, DePaul University

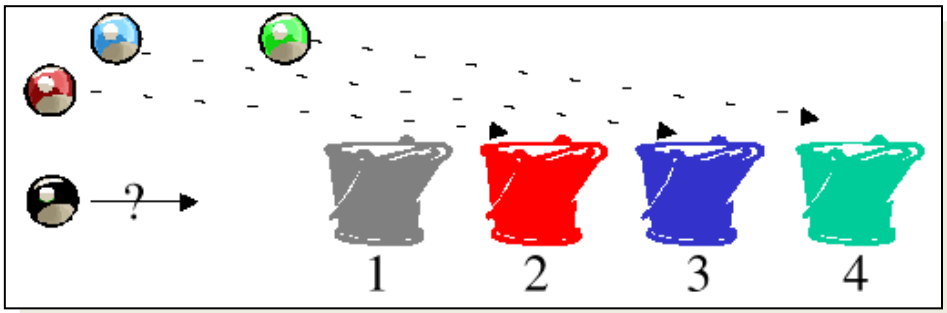
Outline

Basics of Classification

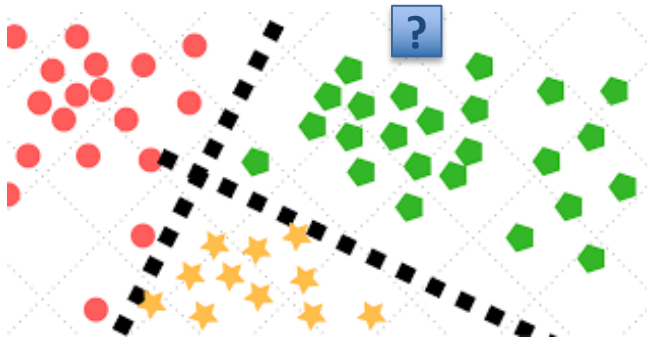
- Definitions and important concepts
- Evaluation

What is classification?

- The goal of data classification is to organize and categorize data in distinct classes
- A model is first created based on the data distribution
- The model is then used to classify new data
- Given the model, a class can be predicted for new data
- Classification = prediction for discrete and nominal values (e.g., class/category labels)



Putting data into buckets



Drawing a boundary between data

https://www.thedataschool.com.au/mohammed-hemayed/explaining-a-classification-model-to-a-client/

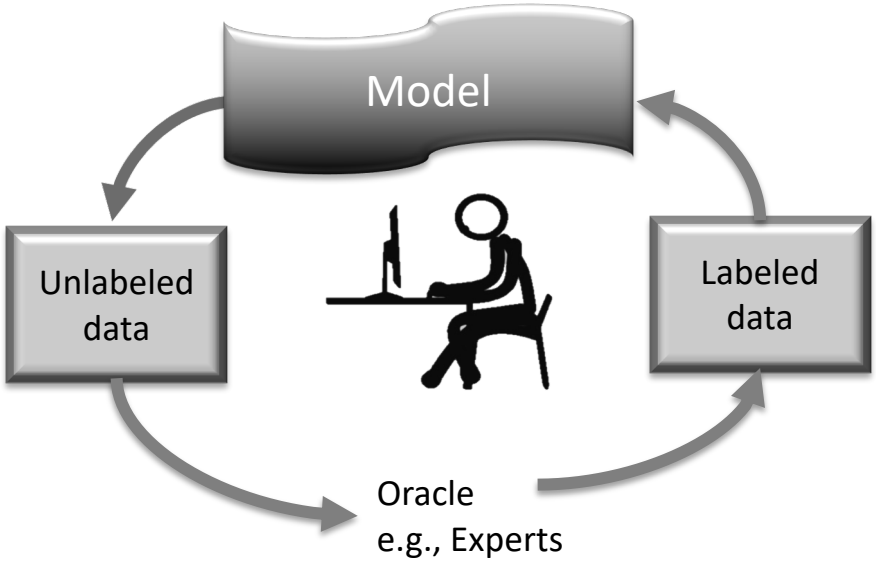
Supervised vs. Unsupervised Learning

- Supervised Learning = Classification
  - **We know the class labels and the number of classes**
- Unsupervised Learning = Clustering
  - **We do not have class labels and may not know the number of groupings in the data**

- **Semi-supervised** Learning: Semi-supervised learning is a class that also make use of unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data.
  - Generative approaches to statistical learning first seek to estimate the distribution of data points belonging to each class
  - Active learning interactively queries the oracle (person/expert or some other information source e.g., expensive experiment) to obtain the desired outputs at new data points

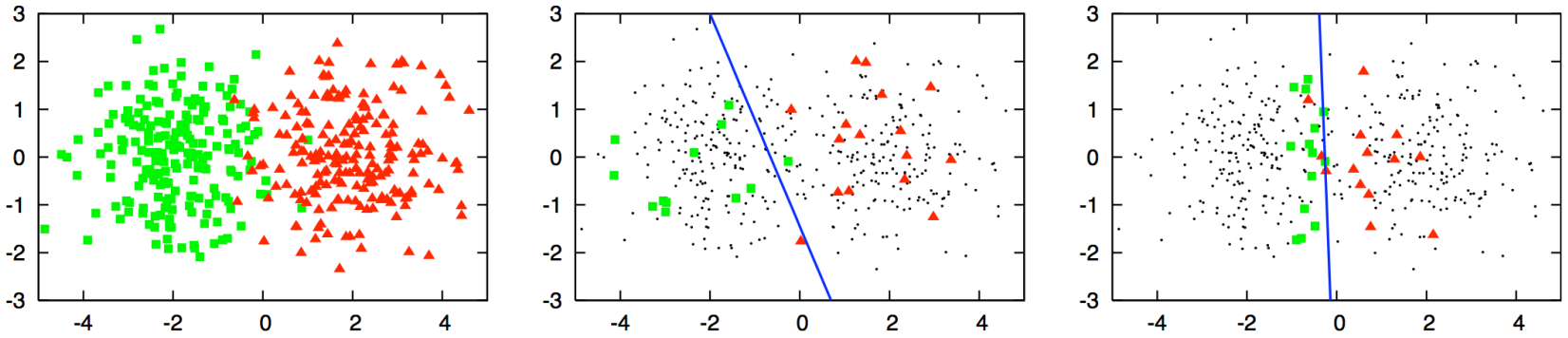
Active Learning

- Build a classifier with a limited number of labels
- Test classifier ➔ poor accuracy
- Request a new data point from the oracle
  - Based on classifier uncertainty (50%/50% for 2 classes)
  - Near the decision boundary



Active Learning

- Build a classifier with a limited number of labels
- Text classifier ➔ poor accuracy
- Request a new data point from the oracle
  - Based on classifier uncertainty (50%/50% for 2 classes)
  - Near the decision boundary
- Accuracy gets better with more points
  - Stop when certain threshold is met



https://www.datacamp.com/community/tutorials/active-learning

Classification: a Formal Definition

- Given:
  - A description of an instance, $x \in X$, where X is the instance or feature space.
    - Typically, x is a row in a table with the instance/feature space described in terms of features or attributes (x is a feature vector)
  - A fixed set of class or category labels: C={c1, c2,…cn}

- Classification task is to determine the class/category of x: $c(x) \in C$, where c(x) is a function whose domain is X and whose range is C.

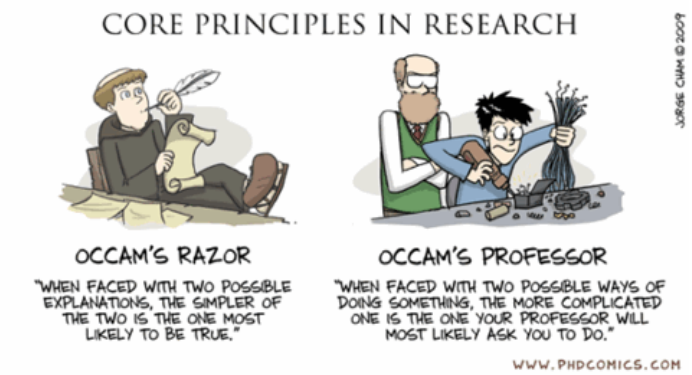
Related Concepts: Lazy vs. Eager

Lazy learner vs. **Eager** Learner

- Lazy learner: (**e.g., KNN**)
  - Data is the model
  - No learning about the underlying structure of the data
  - Leaves all the work for test time, comparing new data point to all previously seen data points
- Eager Learner: (**e.g., Decision trees**)
  - Tries to construct a general, input-independent target function during training
  - Learns a function target as a function(attributes) or a tree, or a probability of classes etc.

Related Concepts: Occam's Razor

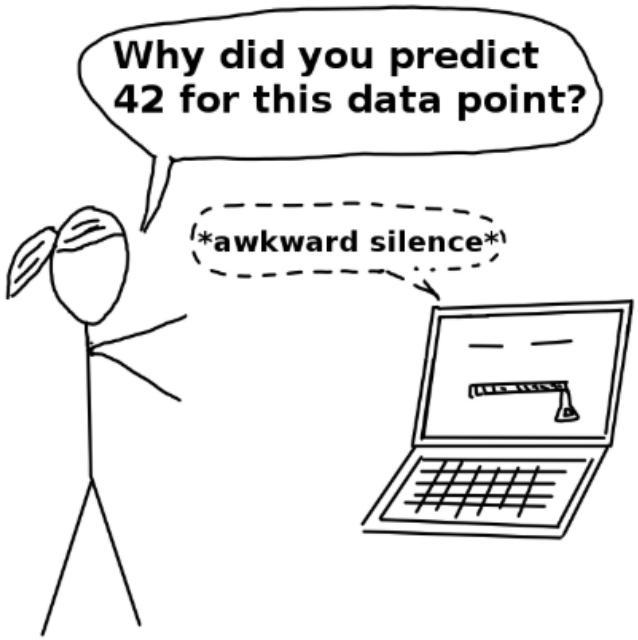
- **Occam's Razor**: based on the idea is attributed to English Franciscan friar William of Ockham (c. 1287–1347) though there are similar formulation before him.
- Occam's Razor summary:
  - Given two models of similar generalization errors, one should prefer the simpler model over the more complex model
  - For **complex models**, there is **a greater chance that it was fitted accidentally by errors in data (overfitting)**
  - Therefore, one should include model complexity when evaluating a model
- Fun Google image search



Related Concepts: **Interpretability**

"It's time to get rid of the black boxes and cultivate trust in Machine Learning"

- **Reliability**
- **Debugging**
- **Informing feature engineering**
- **Directing future data collection**
- **Informing human decision-making**
- **Building Trust**

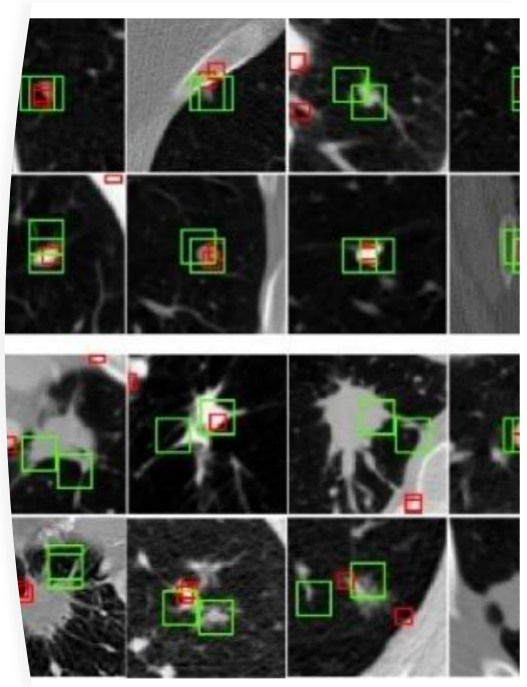


Source: interpretable-ml-book

Related Concepts: **Interpretability**

- Interpretability/explainability is becoming a big topic as data science permeates more fields
  - Goes along with building **trust**.
  - Distinction between the two
    - Establishing cause and effect
    - Understanding the model's results

Visualization to the rescue



Features identified by CNN for positive and negative examples

Qiu, Bowen, et al. "Learning latent spiculated features for lung nodule characterization." *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2020.

- **Black box models:** complex, hard to interpret e.g., deep learning
- **White box models:** explicit, explainable, interpretable; e.g., decision trees
- **Building Trust**

Related Concepts: Labeling

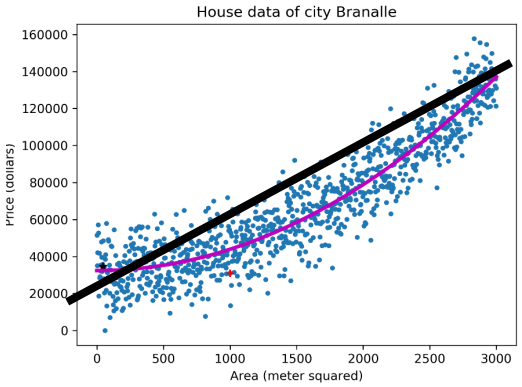
- Target = (Class) Labels = Annotations
- The term **ground truth** refers to the underlying **absolute** state of information; the **gold standard** strives to represent the ground truth as closely as possible (best benchmark/approximation).
- Experiments can be a gold standard though they may not be ground truth (exact equation values)
- Annotations can be a gold standard though they may include human errors (e.g., underlining verbs in sentences to train Natural Language Processing software)
- Labeling can be **crowdsourced** to platforms like **Amazon Turk** and maybe reasonably easy/cheap to obtain with some errors
- Labeling can be **expensive** to obtain from experts or instruments in scientific applications

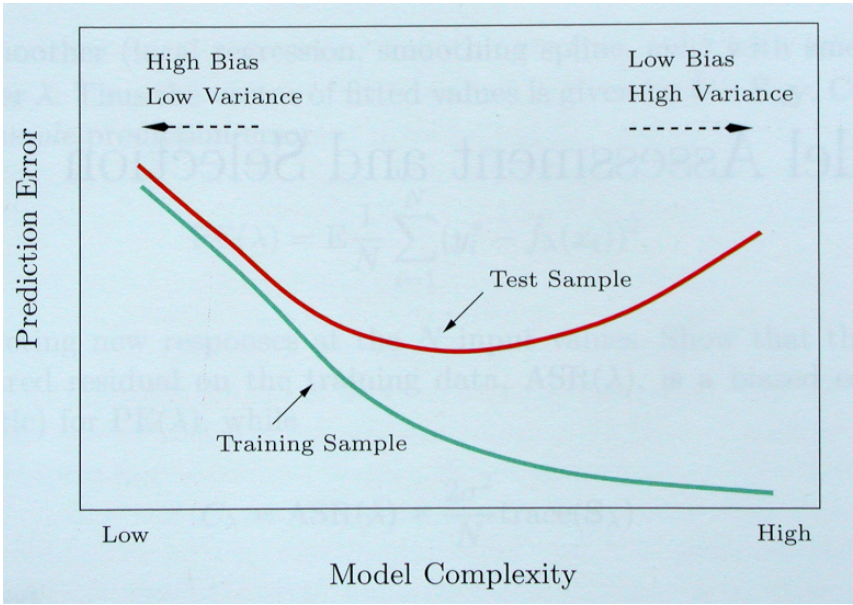
Related Concepts: Bias vs. Variance

- **Bias**: **Any criteria** other than *consistency* with the training data that is used to select a hypothesis (any assumptions on the data)
  - A consistent hypothesis produces results that are consistent with training set
  - Many hypotheses may fit the training data, but a robust model must generalize beyond training data
- **Variance**: is the amount that the estimate of the target function will change if different training data was used.

Related Concepts: Bias vs. Variance

- **Bias**:
  - Low bias: makes no assumptions about the data (structure, distribution etc.) e.g., KNN
  - High bias: makes more assumptions: e.g., Linear regression
- **Variance**:
  - Low variance: small changes to the predictions with changes to the training dataset
  - High variance: large changes to the predictions with changes to the training datasets





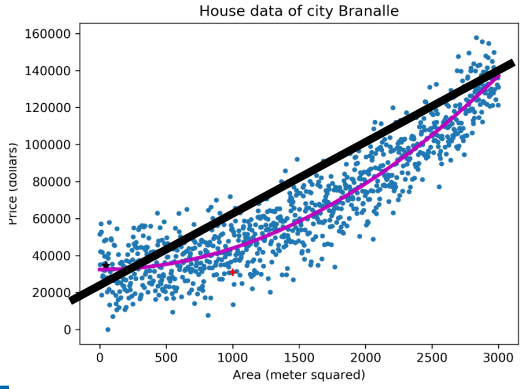
Related Concepts: Bias vs. Variance

Possible ways of dealing with high bias

- Get additional features
- More complex model (e.g., adding polynomial terms such as x_1^2, x_2^2 , $x_1.x_2$, etc.)
- Use smaller regularization coefficient (in regression models)
- **Note:** getting more training data won't necessarily help in this case

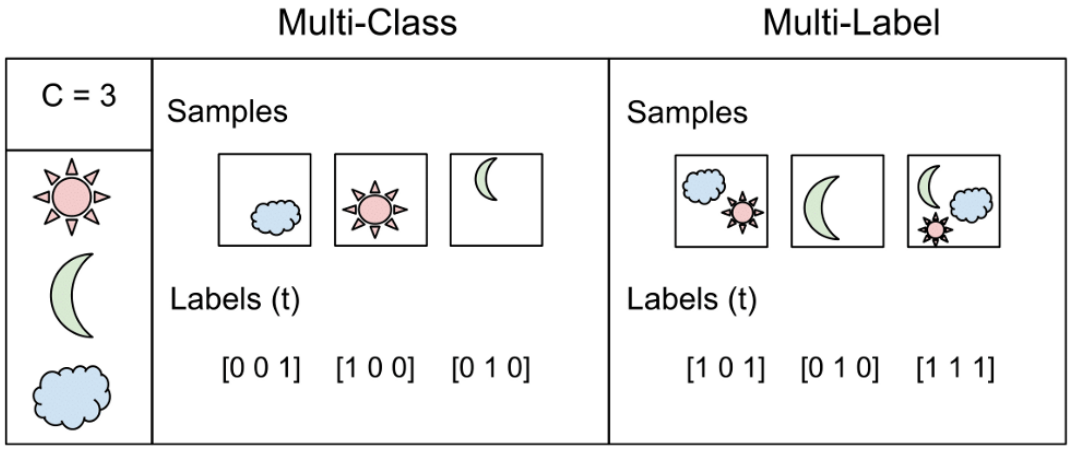
Possible ways dealing with high variance

- Use more training instances
- Reduce the number of features
- Use simpler models
- Use a larger regularization coefficient (in regression models)



Related Concepts: Multi-Class and Multi-Label

- **Multi-label classification**
  - When it is possible to assign multiple classes to an object
  - e.g., it is possible for a movie to be tagged “romance” and “comedy”
- **Multi-class classification**
  - Most of the times, we give examples of binary classification
  - It is possible to have problems where you have three or more classes



Examples:

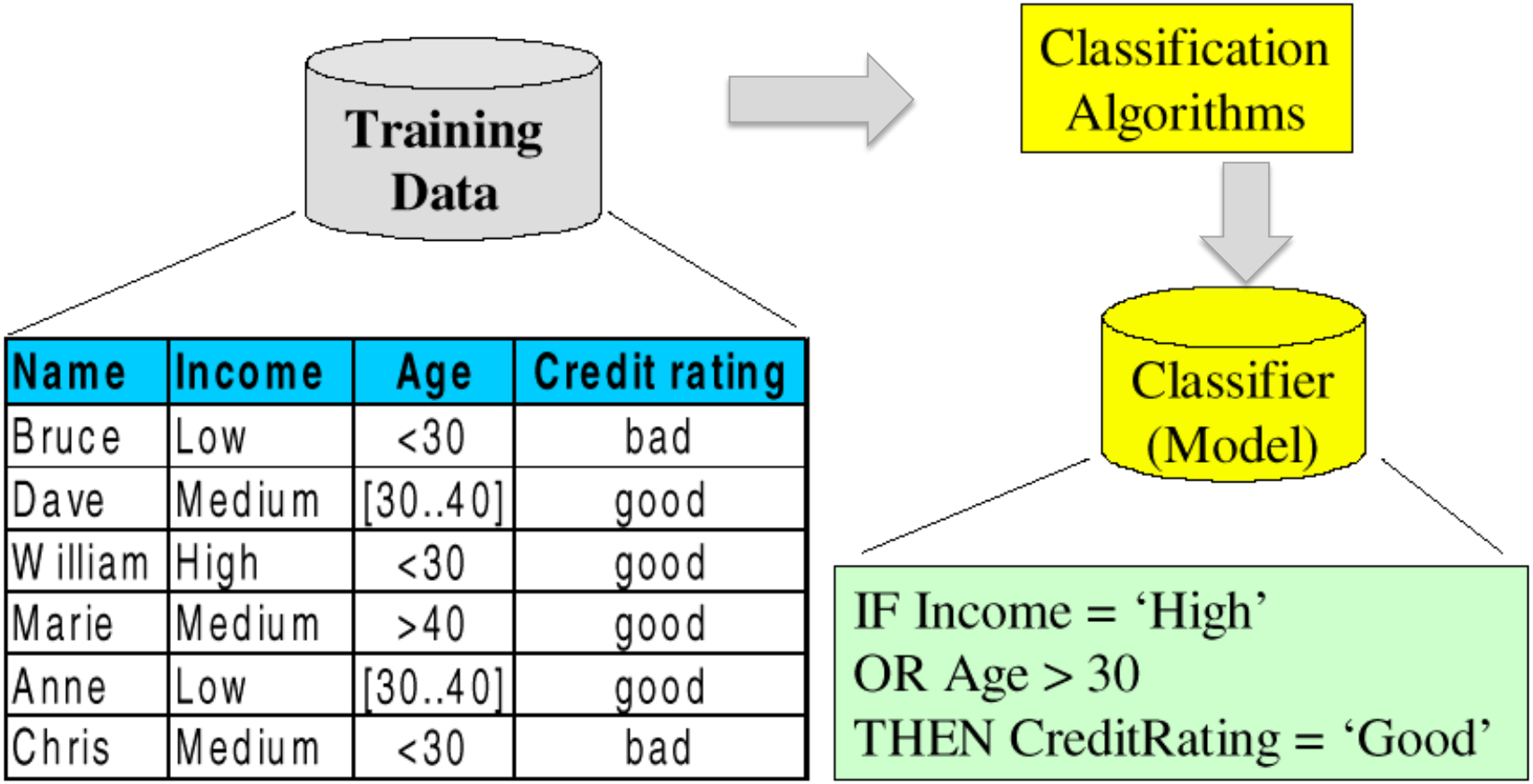
- Movies can be tagged “comedy” and “action”
- Books can be tagged “action” and “thriller”
- etc.

Image source — Google: https://prakhartechviz.blogspot.com/2019/02/multi-label-classification-python.html

Classification: 3 Step Process

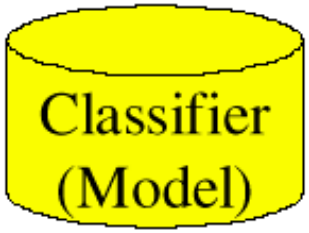
1. Model construction (**Learning**):
   - Each record (instance, example) is assumed to belong to a predefined class, as determined by one of the attributes
     - This attribute is called the target attribute, the class
     - The values of the target attribute are the class labels
   - The set of all instances used for learning the model is called **training set**
   - The model may be represented in many forms: decision trees, probabilities, neural networks, ….
2. Model Evaluation (**Accuracy**):
   - Estimate accuracy rate of the model based on a **test set**
   - The known labels of test instances are compared with the predicts class from model
   - Test set is independent of training set otherwise over-fitting will occur
3. Model Use (**Classification**):
   - The model is used to classify unseen instances (**evaluation set**) (i.e., to predict the class labels for new unclassified instances)
   - Predict the value of an actual attribute

Step 1: Model Construction



Name	Income	Age	Credit rating
Bruce	Low	<30	bad
Dave	Medium	[30..40]	good
William	High	<30	good
Marie	Medium	>40	good
Anne	Low	[30..40]	good
Chris	Medium	<30	bad

Step 2: Model Evaluation

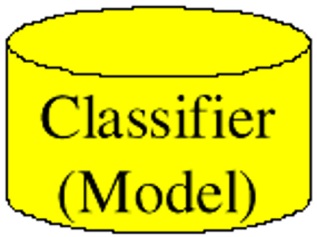


Name	Income	Age	Credit rating
Tom	Medium	<30	bad
Jane	High	<30	bad
Wei	High	>40	good
Hua	Medium	[30..40]	good

How accurate is the model?

IF Income = 'High'
OR Age > 30
THEN CreditRating = 'Good'

Step 3: Model Use



Name	Income	Age	Credit rating
Paul	High	[30..40]	?

Credit Rating?

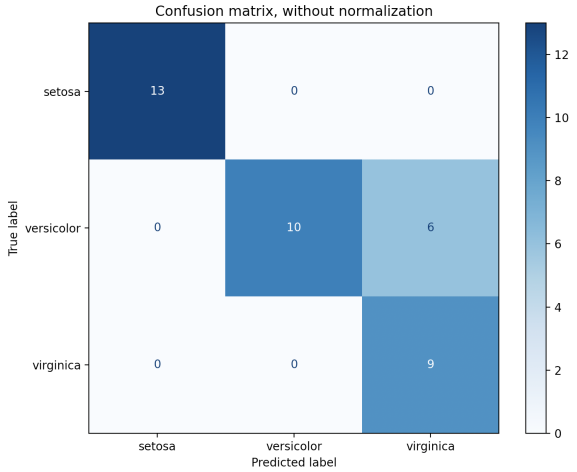
good

Classification: Data for 3 Step Process

- To train and evaluate models, data are often divided into three sets: the **training set**, the **test set**, and **the evaluation set**
- Training Set
  - is used to build the initial model
  - may need to "**enrich the data**" to get enough of the special (corner) cases
- Test Set
  - is used to adjust the initial model
  - models can be tweaked to be less idiosyncrasies to the training data and can be adapted for a more general model
  - idea is to prevent **overfitting** (i.e., finding patterns where none exist).
- Evaluation Set is used to evaluate the final model performance

Classification: Basic Evaluation

- Confusion matrix tabulates correct and incorrect classification based on class labels
  - Compares predictions to true labels
  - Correct predictions are along the diagonal
  - Errors are in other cells



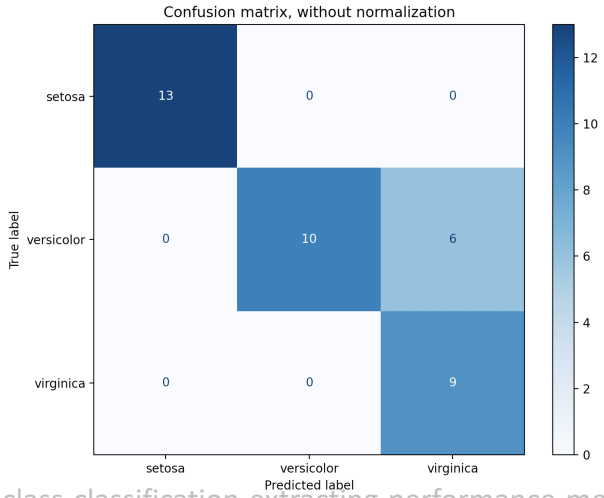
Classification: Basic Evaluation

- Confusion matrix tabulates correct and incorrect classification based on class labels
- In the binary case:
  - TP: True label +, classified +
  - FP: True label -, classified +
  - TN: True label - , classified -
  - FN: True label +, classified -

Classification: Basic Evaluation Metric

Classifier Accuracy, or recognition rate: percentage of test set instances that are correctly classified

- **Accuracy** = (total correct predictions)/All
- **Accuracy (binary)** = (TP + TN)/All
- **Error rate**: 1 – Accuracy, or Error rate = (FP + FN)/All



https://towardsdatascience.com/multi-class-classification-extracting-performance-metrics-from-the-confusion-matrix-b379b427a872

Classification: Basic Evaluation Metric

- When the output field is nominal (e.g., in two-class prediction), we use a confusion matrix to evaluate the resulting model

- Example

Predicted Class

Actual Class	T	F	Total
T	18	2	20
F	3	15	18
Total	21	17	38

  - Overall correct classification rate = (18 + 15) / 38 = 87%
  - Given T, correct classification rate = 18 / 20 = 90% (TPR, Recall, Sensitivity)
  - Given F, correct classification rate = 15 / 18 = 83% (TNR, Specificity)

Other Evaluation Metrics

- Precision
  - % of instances that the classifier predicted as positive that are actually positive

$$precision = \frac{TP}{TP + FP}$$

- Recall
  - % of positive instances that the classifier predicted correctly as positive
  - a.k.a "Completeness"

$$recall = \frac{TP}{TP + FN}$$

- Perfect score for both is 1.0, but there is often a trade-off between Precision and Recall

- *F* measure (F_1 or *F*-score)
  - harmonic mean of precision and recall

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

Evaluation: Overfitting

- **Overfitting**: model is too complex, reads too much into the training set
  - model is trying too hard as to learn noise in training data or find patterns when there are none
  - common problem with most data mining algorithms
  - resulting model works well on the training set but poorly on unseen data
  - Need to make sure test set is independent of training set
  - Check for assumptions
  - Use robust evaluation methods
- **Underfitting**: model is too simple, neither fits training nor test data
- Concept is related to bias-variance tradeoff though a model can overfit without being complex
  - e.g., too many features
  - Not enough data, or biased data

Evaluation Issues and Methods

- Insufficient data to divide into three disjoint sets?
  - In such cases, validation techniques can play a major role
    - Cross Validation
    - Bootstrap Validation
- Class imbalance: Need to make sure to work with representative samples

Cross Validation

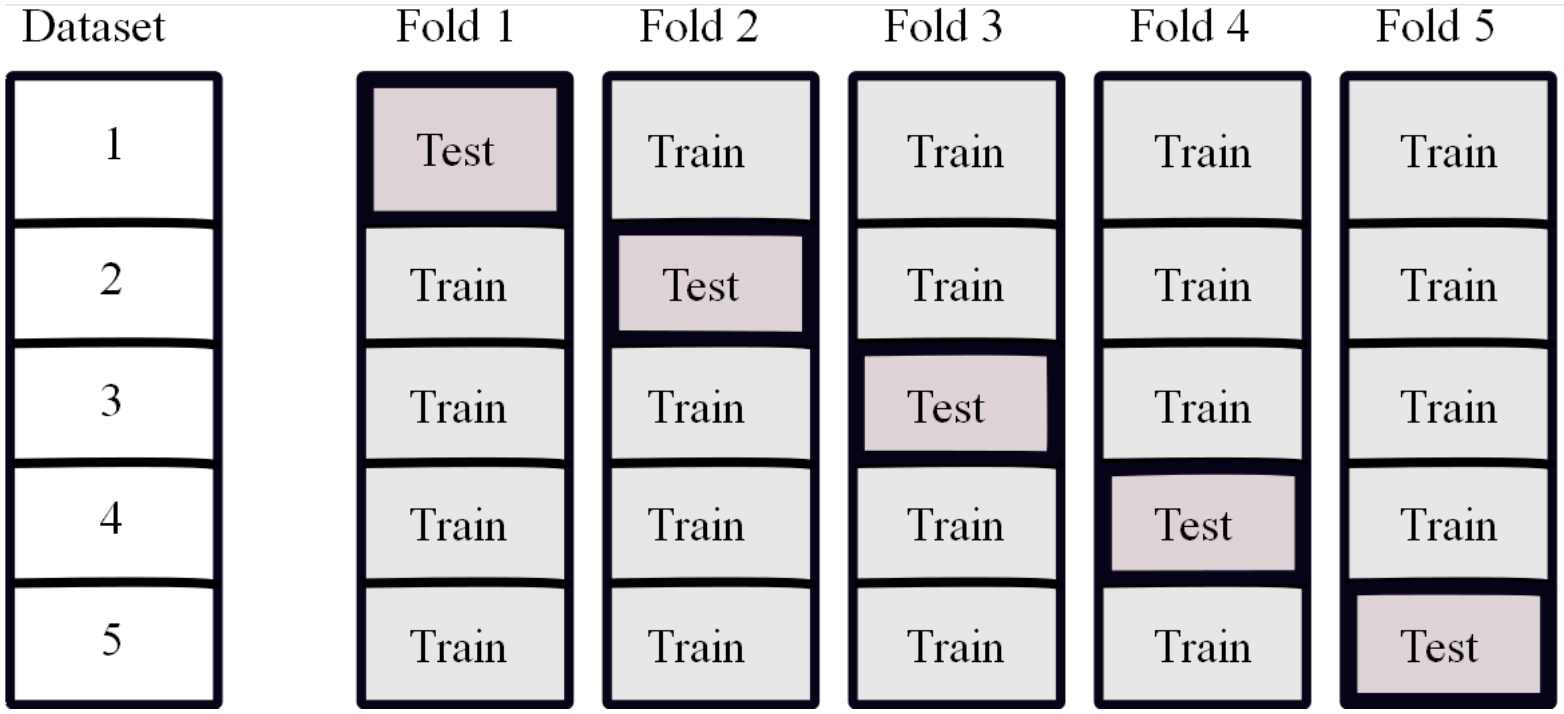
- Randomly divide the data into **n folds**, each with approximately the same number of records
- Create n models using the same algorithms and training parameters; each model is trained with n-1 folds of the data and tested on the remaining fold
- Can be used to find the best algorithm and its optimal training parameter

Cross Validation Steps

1. Divide the available data into a training set and an evaluation set
2. Split the training data into *n* folds
3. Select an algorithm and training parameters
4. Train and test *n* models using the *n* train-test splits
5. Repeat step 2 to 4 using different algorithms / parameters and compare model accuracies
6. Select the ***best*** combination (of algorithms and parameters)
7. Use all the training data to train the final model – **Get an averaged validation score**
8. Assess the final model using the evaluation set

5-Fold Cross Validation

Similar to several train/test split except, you make sure you have used every point in data in training and testing at least once



How many K?

- No free lunch
- Popular values 5, 10

Evaluation: Class Imbalance & Stratified Sampling and cross validation

- Your data objects may be divided into separate "stratum"
- Some classes/strata may by over-represented and random sampling would ignore important "minority examples"
- Use stratified sampling and stratified cross-validation

Bootstrap Validation

- Based on the statistical procedure of sampling with replacement
  - data set of *n* instances is sampled n times (with replacement) to give another data set of *n* instances
  - since some elements will be repeated, there will be elements in the original data set that are not picked
  - these remaining instances are used as the test set
- How many instances in the test set?
  - Probability of not getting picked in one sampling = $1 - 1/n$
  - Pr(not getting picked in *n* samples) = $(1 - 1/n)^n = e^{-1} = 0.368$
  - so, for large data set, test set will contain about 36.8% of instances
  - to compensate for smaller training sample (63.2%), test set error rate is combined with the re-substitution error in training set:

$$e = (0.632 * e_{\text{test instance}}) + (0.368 * e_{\text{training instance}})$$

Bootstrap Validation

Bootstrap Validation

- Bootstrap validation is used in ensemble learning
- Different models are trained with different samples
- The ensemble reports an “aggregate” accuracy based on the accuracy of the individual models
- We revisit this topic later in the class

Learning issues

- Simplicity and Interpretability
- Bias vs. variance
- **Overfitting**
- **Accuracy** (% of instances classified correctly) measured on **independent** test data
  - Evaluation metrics
  - Evaluation methods
  - Enough data
  - Too many features
  - **Class imbalance**
- Efficiency Issues:
  - Training time (efficiency of training algorithm)
  - Testing time (efficiency of subsequent classification)

Example Classifiers

- Decision Tree Induction
- Bayesian Classification
- K-Nearest Neighbor
- Neural Networks
- Linear Discriminant Analysis
- Support Vector Machines
- Genetic Algorithms
- Many More ….

- Also, Ensemble Methods, e.g., Random Forest