THE UNIVERSITY OF WESTERN AUSTRALIA

School of Computer Science and Software Engineering

CITS4009 Introduction to Data Science

SEMESTER 2, 2017: PART 2 MODELLING METHODS

CHAPTER 5 CHOOSING AND EVALUATING MODELS

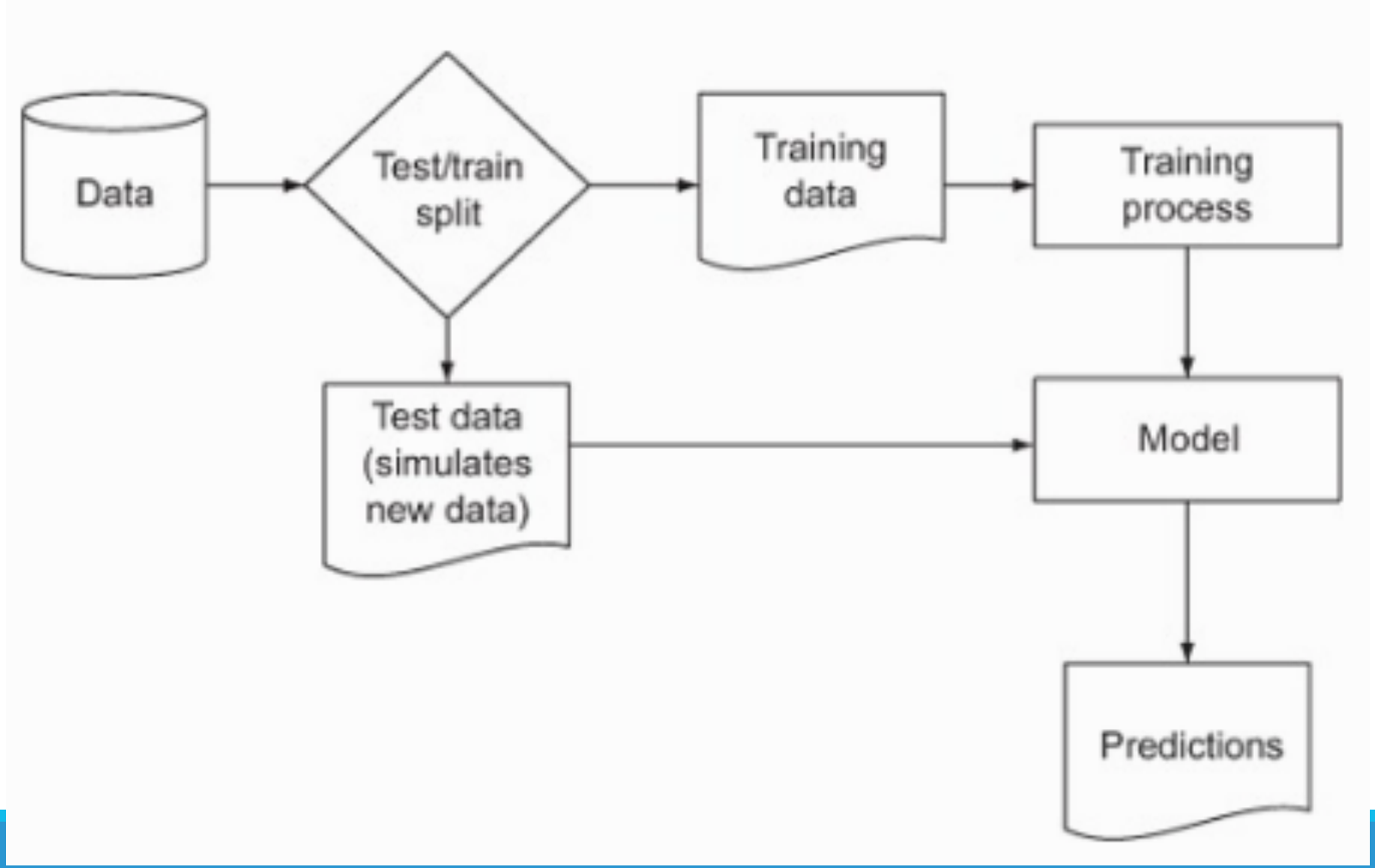
Chapter Objectives

- Mapping business problems to machine learning tasks
- Evaluating model quality
- Validating model soundness

introduction

- Data Scientist's ultimate goal is to solve a concrete business problem:
  - Increase look-to-buy ratio,
  - identify fraudulent transactions,
  - predict and manage the losses of a loan portfolio, and so on.
- Statistical modeling used to solve any given problem.
- ***Model evaluation*** defined as quantifying the performance of a model.
- ***Model validation*** defined as the generation of an assurance that the model will work in production as well as it worked during training.

Schematic model construction and evaluation



Model

- It is a disaster to build a model that works great on the original training data and then performs poorly when used in production
- The biggest cause of model validation failures is not having enough training data to represent the variety of what may later be encountered in production.

Mapping problems to machine learning tasks

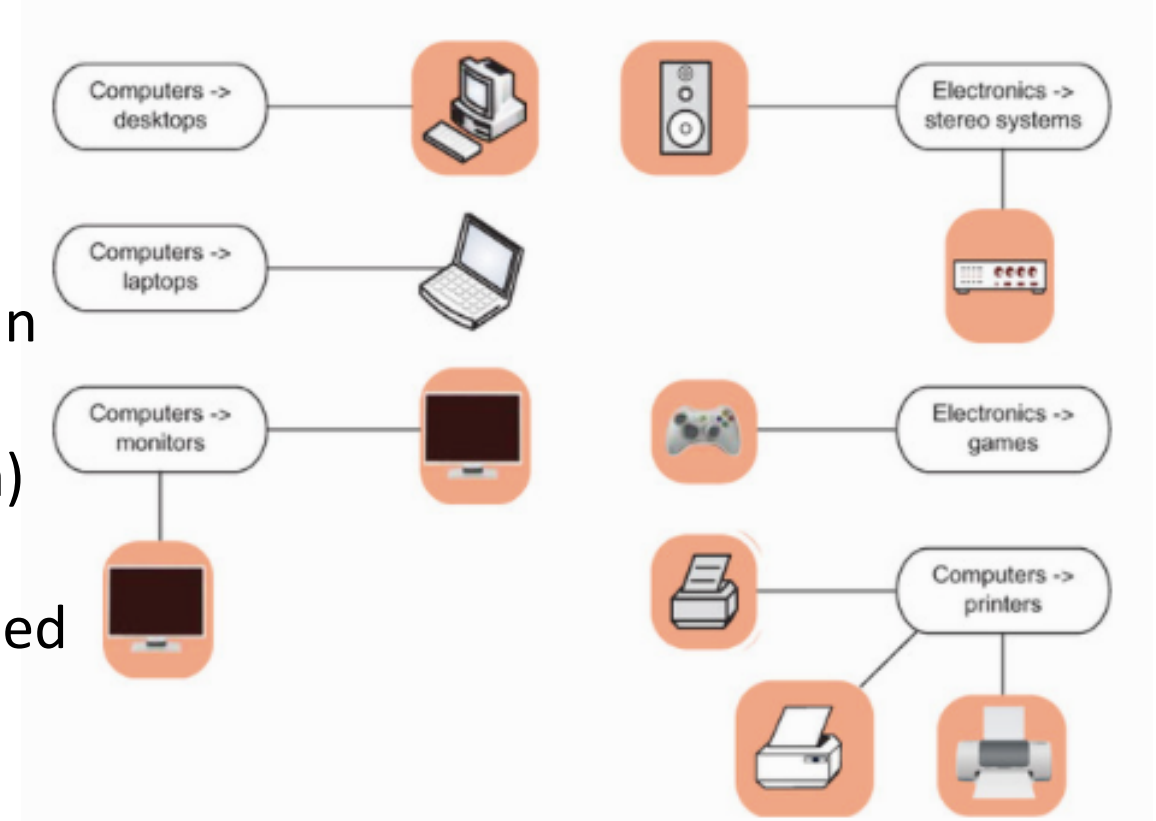
- Map a business problem to a good machine learning method
- Example of problems in an online retail company:
  - Predicting what customers might buy, based on past transactions
  - Identifying fraudulent transactions
  - Determining price elasticity (the rate at which a price increase will decrease sales, and vice versa) of various products or product classes
  - Determining the best way to present product listings when a customer searches for an item
  - Customer segmentation: grouping customers with similar purchasing behavior
  - AdWord valuation: how much the company should spend to buy certain AdWords on search engines
  - Evaluating marketing campaigns
- **What methods you should use?**

Solving classification problems

Your task is to automate the assignment of new products to your company's product categories:

1. It is a difficult task.
2. Many large online retailers use teams of human taggers to hand-categorize their products.

- Classification is deciding how to assign (known) labels to an object.
- Classification itself is an example of what is called supervised learning.
- Building training data is the major expense for most classification tasks, especially text-related ones.



Assigning products to product categories

Some common classification methods

Method	Description
Naive Bayes	Naive Bayes classifiers are especially useful for problems with many input variables, categorical input variables with a very large number of possible values, and text classification. Naive Bayes would be a good first attempt at solving the product categorization problem.
Decision trees	Decision trees are useful when input variables interact with the output ”if-then” kinds of ways (such as IF age > 65, THEN has.health.insurance=T). They are also suitable when inputs have an AND relationship to each other (such as IF age < 25 AND student=T, THEN...) or when input variables are redundant or correlated. The decision rules that come from a decision tree are in principle easier for nontechnical users to understand than the decision processes that come from other classifiers. In section 6.3.2, we’ll discuss an important extension of decision trees: random forests.

Some common classification methods-cont.

Method	Description
Logistic Regression	Logistic regression is appropriate when you want to estimate class probabilities (the probability that an object is in a given class) in addition to class assignments.[a] An example use of a logistic regression–based classifier is estimating the probability of fraud in credit card purchases. Logistic regression is also a good choice when you want an idea of the relative impact of different input variables on the output. For example, you might find out that a $100 increase in transaction size increases the odds that the transaction is fraud by 2%, all else being equal.
Support vector machines	Support vector machines (SVMs) are useful when there are very many input variables or when input variables interact with the outcome or with each other in complicated (nonlinear) ways. SVMs make fewer assumptions about variable distribution than do many other methods, which makes them especially useful when the training data isn't completely representative of the way the data is distributed in production.

Multicategory vs. two-category classification

- Product classification is an example of multicategory or multinomial classification
- Most classification problems and most classification algorithms are specialized for two-category, or binomial, classification.
- building one classifier for each category, called a one versus rest" classifier
- multiple-category implementation, as they tend to work better than multiple binary classifiers

Solving scoring problems

- Suppose that your task is to help evaluate how different marketing campaigns can increase valuable traffic to the website. The goal is not only to bring more people to the site, but to bring more people who buy
- the communication channel (ads on websites, YouTube videos, print media, email, and so on);
- the traffic source (Facebook, Google, radio stations, and so on); the demographic targeted; the time of year;
- Predicting the increase in sales from a particular marketing campaign is an example of regression, or scoring
- Scoring is an instance of supervised learning.

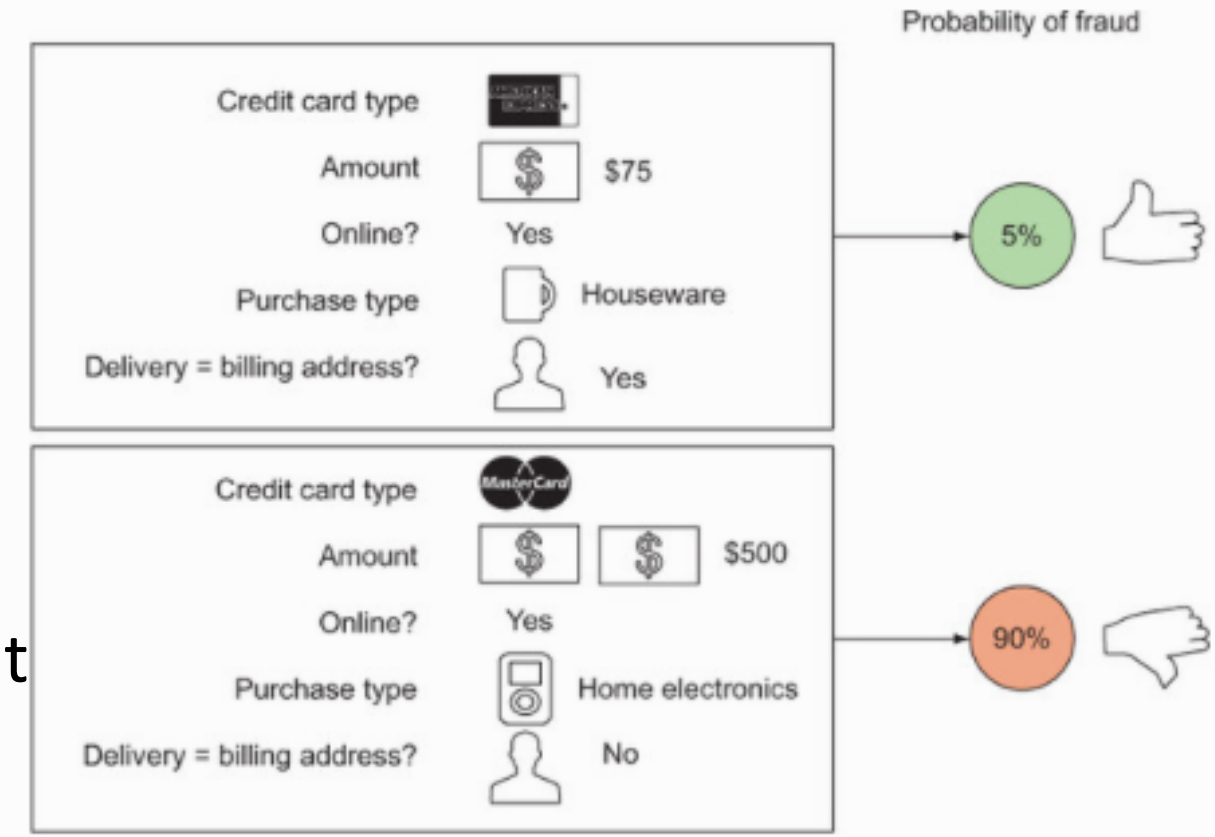
Common Scoring Methods

Liner Regression

Builds a model such that the predicted numerical output is a linear additive function of the inputs. This can be a very effective approximation, even when the underlying situation is in fact nonlinear.

Logistic Regression

Predicts a value between 0 and 1, making it suitable for predicting probabilities (when the observed outcome is a categorical value) and rates (when the observed outcome is a rate or ratio).



Notional example of determining the probability that a transaction is fraudulent.

Working without known targets

- You need to have a training dataset,
- You may be looking for patterns and relationships in the data that will help you understand your customers or your business better.
- **Unsupervised learning**: rather than predicting outputs based on inputs, the objective of unsupervised learning is to discover similarities and relationships in the data.
- Some common clustering methods include these:
  - K-means clustering
  - Apriori algorithm for finding association rules
  - Nearest neighbor

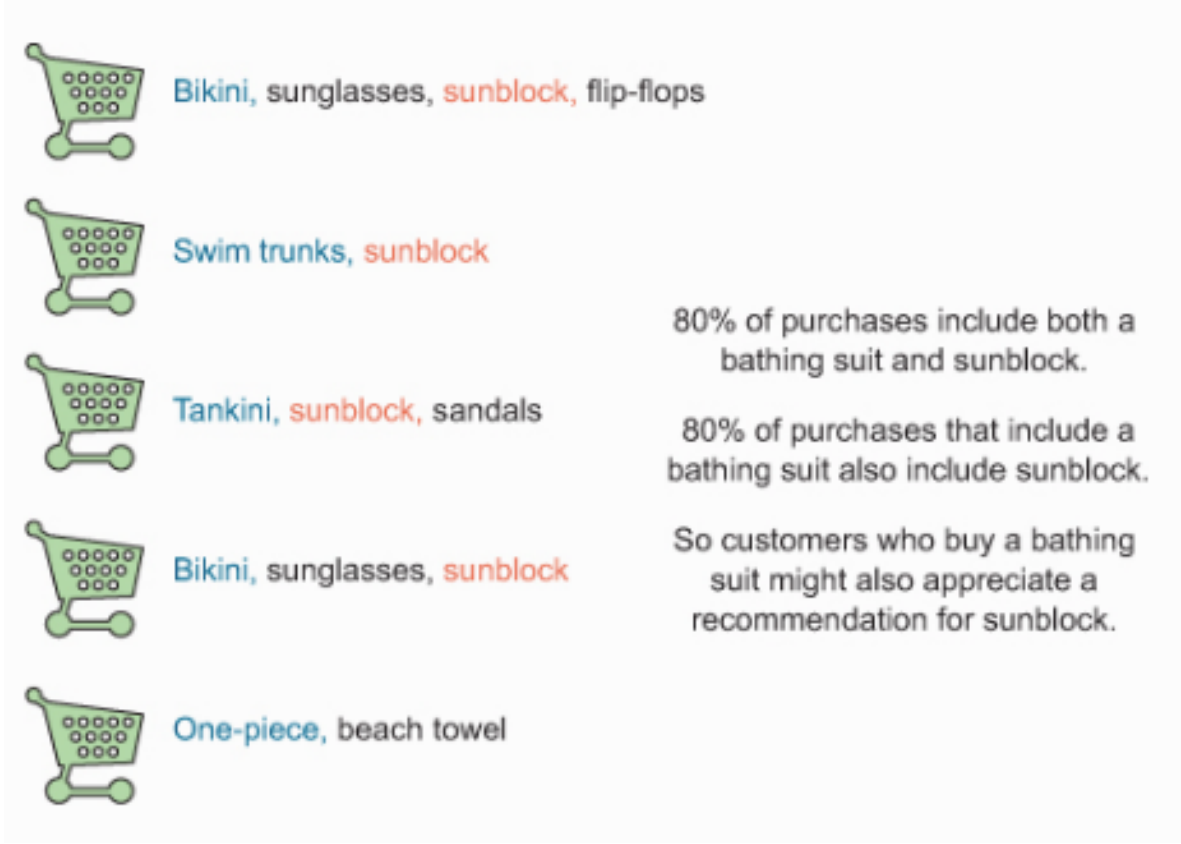
When to use basic clustering

- K-means clustering is one way to sort the data into groups such that members of a cluster are more similar to each other than they are to members of other clusters.

Notional example of clustering your customers by purchase pattern and purchase amount

When to use association rules

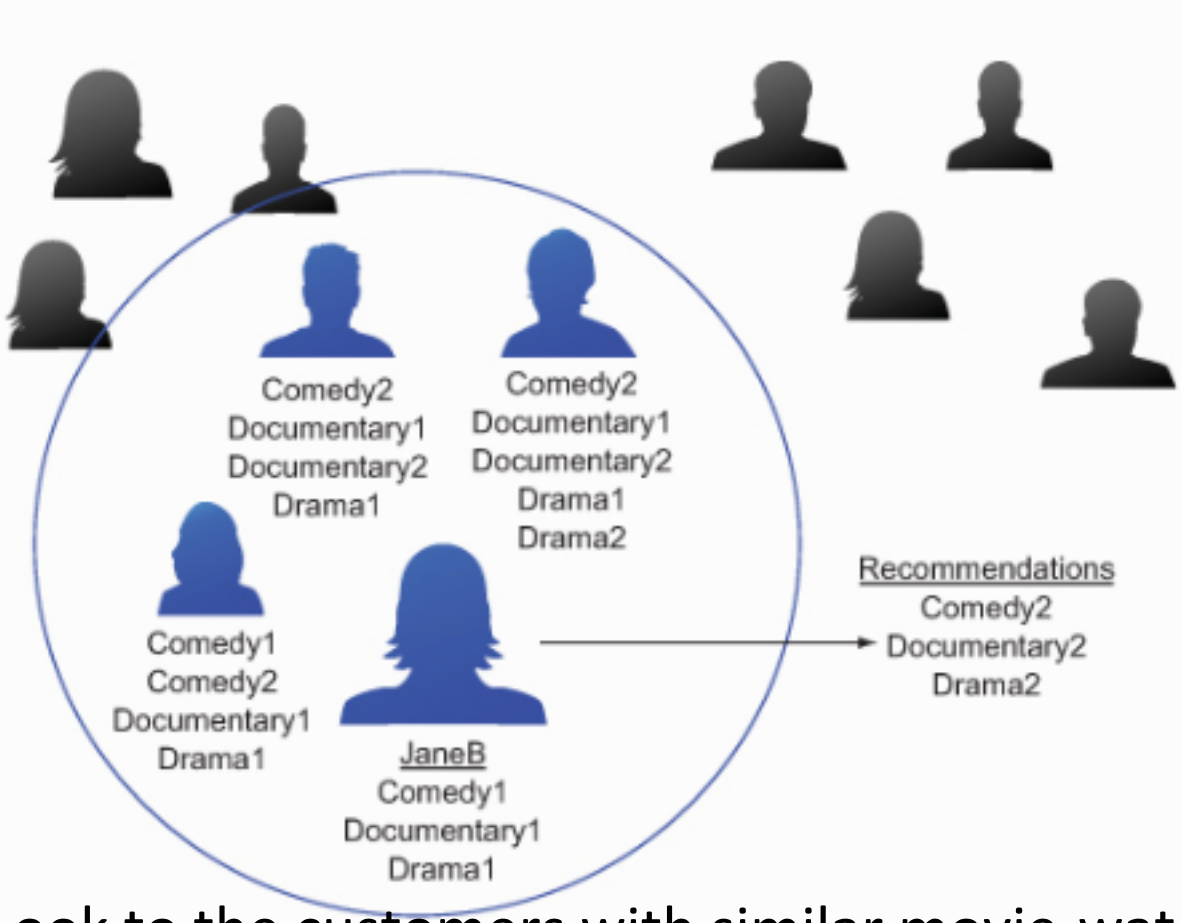
which products tend to be purchased together.
mine useful product recommendations: whenever you observe that someone has put a bathing suit into their shopping cart, you can recommend suntan lotion, as well.



Notional example of finding purchase patterns in your data

When to use nearest neighbor methods

- Find similarities in people to make product recommendation
- This can be solved with nearest neighbor (or k-nearest neighbor methods, with K = 3)
- Nearest neighbor algorithms predict something about a data point *p* (like a customer's future purchases) based on the data point or points that are most similar to *p*.



Look to the customers with similar movie-watch patterns as JaneB for her movie recommendatio

Problem-to-method mapping

Example tasks	Machine learning terminology	Typical algorithms
Identifying spam email Sorting products in a product catalog Identifying loans that are about to default Assigning customers to customer cluster	Classification: assigning known labels to objects	Decision trees Naive Bayes Logistic regression (with a threshold) Support vector machines
Predicting the value of AdWords Estimating the probability that a loan will default Predicting how much a marketing campaign will increase traffic or sales	Regression: predicting or forecasting numerical values	Linear regression Logistic regression

Example tasks	Machine learning terminology	Typical algorithms
Finding products that are purchased together Identifying web pages that are often visited in the same session Identifying successful (much-clicked) combinations of web pages and AdWords	Association rules: finding objects that tend to appear in the data together finding objects that tend to appear in the data together	Apriori
Identifying groups of customers with the same buying patterns Identifying groups of products that are popular in the same regions or with the same customer clusters Identifying news items that are all discussing similar events	Clustering: finding groups of objects that are more similar to each other than to objects in other groups	K-means
Making product recommendations for a customer based on the purchases of other similar customers Predicting the final price of an auction item based on the final prices of similar products that have been	Nearest neighbor: predicting a property of a datum based on the datum or data that are most similar to it	Nearest neighbor

Example tasks	Machine learning terminology	Typical algorithms
Making product recommendations for a customer based on the purchases of other similar customers Predicting the final price of an auction item based on the final prices of similar products that have been auctioned in the past	Nearest neighbor: predicting a property of a datum based on the datum or data that are most similar to it	Nearest neighbor

Prediction vs. forecasting

- To predict is to pick an outcome, such as It will rain tomorrow," and to forecast is to assign a probability: There's an 80% chance it will rain tomorrow.

Evaluating models

- First thing to check is if the model even works on the data it was trained from.
- Evaluation model:
  - Classification
  - Scoring
  - Probability estimation
  - Ranking
  - Clustering
- Compute one or two summary scores that tell if the model is effective.

To decide if a given score is high or low

Ideal model	Purpose
Null model	A null model is the best model of a very simple form you're trying to outperform. The two most typical null model choices are a model that is a single constant (returns the same answer for all situations) or a model that is independent (doesn't record any important relation or interaction between inputs and outputs). We use null models to lower-bound desired performance, so we usually compare to a best null model. For example, in a categorical problem, the null model would always return the most popular category (as this is the easy guess that is least often wrong); for a score model, the null model is often the average of all the outcomes (as this has the least square deviation from all of the outcomes); and so on. The idea is this: if you're not out-performing the null model, you're not delivering value. Note that it can be hard to do as good as the best null model, because even though the null model is simple, it's privileged to know the overall distribution of the items it will be quizzed on. We always assume the null model we're comparing to is the best of all possible null models.

To decide if a given score is high or low

Ideal model	Purpose
A Bayes rate model	A Bayes rate model (also sometimes called a saturated model) is a best possible model given the data at hand. The Bayes rate model is the perfect model and it only makes mistakes when there are multiple examples with the exact same set of known facts (same xs) but different outcomes (different ys). It isn't always practical to construct the Bayes rate model, but we invoke it as an upper bound on a model evaluation score. If we feel our model is performing significantly above the null model rate and is approaching the Bayes rate, then we can stop tuning. When we have a lot of data and very few modeling features, we can estimate the Bayes error rate. Another way to estimate the Bayes rate is to ask several different people to score the same small sample of your data; the found inconsistency rate can be an estimate of the Bayes rate.

To decide if a given score is high or low

Ideal model	Purpose
Single-variable models	We also suggest comparing any complicated model against the best single-variable model you have available. A complicated model can't be justified if it doesn't outperform the best single-variable model available from your training data. Also, business analysts have many tools for building effective single-variable models (such as pivot tables), so if your client is an analyst, they're likely looking for performance above this level.

Evaluating classification models

- The most common measure of classifier quality is accuracy
- confusion matrix and show how it can be used to calculate many important evaluation scores.
- The confusion matrix is a table counting how often each combination of known outcomes (the truth) occurred in combination with each prediction type

Confusion matrix

The absolute most interesting summary of classifier performance is the confusion matrix.

This matrix is just a table that summarizes the classifier's predictions against the actual known data categories.

The confusion matrix is a table counting how often each combination of known outcomes (the truth) occurred in combination with each prediction type.

Confusion matrix

For our email spam example, the confusion matrix is given by

>cM-<table(truth=spamTest$spam,prediction=spamTest$pred>0.5)

>print(cM)

Prediction

truth FALSE TRUE

Non-spam 264 14

Spam 22 158

Using this summary, we can now start to judge the performance of the model.

Accuracy

- Accuracy is defined as the number of items categorized correctly divided by the total number of items.
- It’s simply what fraction of the time the classifier is correct.

Categorization accuracy isn't the same as numeric accuracy

- **Categorization accuracy isn't the same as numeric accuracy**

It's important to not confuse accuracy used in a classification sense with accuracy used in a numeric sense

- **Accuracy is an inappropriate measure for unbalanced classes**

Precision and recall

- Precision is what fraction of the items the classifier flags as being in the class actually are in the class.
- precision is how often a positive indication turns out to be correct. It's important to remember that precision is a function of the combination of the classifier and the dataset. It doesn't make sense to ask how precise a classifier is in isolation; it's only sensible to ask how precise a classifier is for a given dataset.
- Precision is a measure of confirmation (when the classifier indicates positive, how often it is in fact correct),
- Recall is a measure of utility (how much the classifier finds of what there actually is to find).

F1

- F1 score is a useful combination of precision and recall. If either precision or recall is very small, then F1 is also very small. F1 is defined as `2*precision*recall/(precision+recall)`.

Sensitivity and specificity

- Sensitivity is also called the true positive rate and is exactly equal to recall. Specificity is also called the true negative rate and is equal to

```
TN/(TN+FP)=cM[1,1]/(cM[1,1] +cM[1,2])
```

- null classifiers (classifiers that always say positive or always say negative) always return a zero score on either sensitivity or specificity.

Common classification performance measures

Measure	Formula
Accuracy	TP+TN)/(TP+FP+TN+FN)
Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
Sensitivity	TP/(TP+FN)
Specificity	TN/(TN+FP)

The main idea is to use these standard scores and then work with your client and sponsor to see what most models their business needs

Classifier performance measures business stories

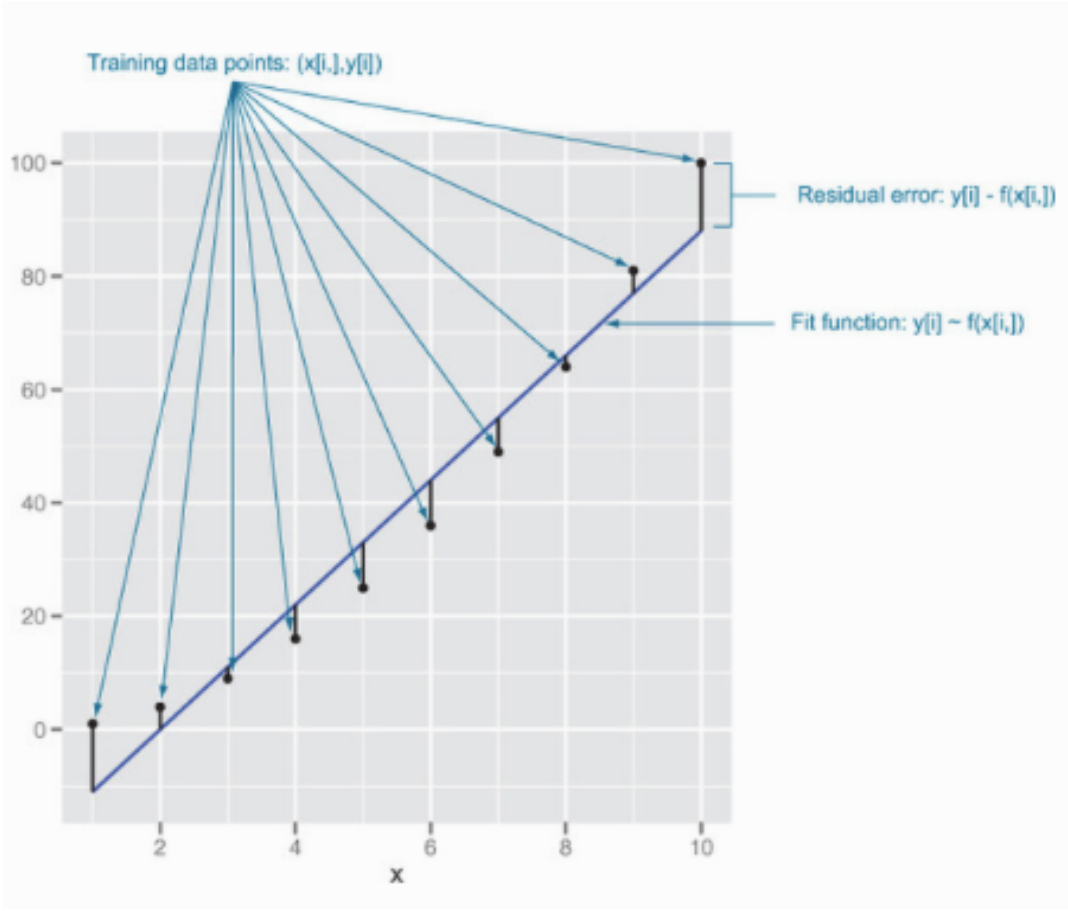
Measure	Typical business need	Follow-up question
Accuracy	We need most of our decisions to be correct.	Can we tolerate being wrong 5% of the time? And do users see mistakes like spam marked as non-spam or non-spam marked as spam as being equivalent?
Precision	Most of what we marked as spam had darn well better be spam.	That would guarantee that most of what is in the spam folder is in fact spam, but it isn't the best way to measure what fraction of the user's legitimate email is lost. We could cheat on this goal by sending all our users a bunch of easy-to-identify spam that we correctly identify. Maybe we really want good specificity.

Classifier performance measures business stories

Measure	Typical business need	Follow-up question
Recall	We want to cut down on the amount of spam a user sees by a factor of 10 (eliminate 90% of the spam).	If 10% of the spam gets through, will the user see mostly non-spam mail or mostly spam? Will this result in a good user experience?
Sensitivity	We have to cut a lot of spam, other wise the user won't see a benefit.	If we cut spam down to 1% of what it is now, would of what it is now, would that be a good user experience?
Specificity	We must be at least three nines on legitimate email; the user must see at least 99.9% of their non-spam email.	Will the user tolerate missing 0.1% of their legitimate email, and should we keep a spam folder the user can look at?

Evaluating scoring models

- The main concept is looking at what is called the residuals or the difference between our predictions f(x[i,]) and actual outcomes y[i].



Root mean square error

- This is the square root of the average square of the difference between our prediction and actual values.
- Think of it as being like a standard deviation: how much your prediction is typically off.

R-squared

- or R^2, or the coefficient of determination
- It's defined as 1.0 minus how much unexplained variance your model leaves (measured relative to a null model of just using the average y as a prediction)
- R-squared is dimensionless (it's not the units of what you're trying to predict), and the best possible R-squared is 1.0 (with near-zero or negative R-squared being horrible).
- R-squared is equal to the square of another measure called the correlation (or Pearson product-moment correlation coefficient;

R-squared

- R-squared is not always the best business-oriented metric.
- For example, it’s hard to tell what a 10% reduction of RMSE would mean in relation to the Netflix Prize. But it would be easy to map the number of ranking errors and amount of suggestion diversity to actual Netflix business benefits.

Correlation

- Correlation is very helpful in checking if variables are potentially useful in a model
- Three calculations that go by the name of correlation:
  1. Pearson coefficient checks for linear relations,
  2. Spearman coefficient checks for rank or ordered relations
  3. Kendall coefficient checks for degree of voting agreement

Don’t use correlation to evaluate model quality in production

- It is advised to not use the correlation to measure model quality because:
- correlation ignores shifts and scaling factors. So correlation is actually computing if there is any shift and rescaling of your predictor that is a good predictor.
- This isn’t a problem for training data (as these predictions tend to not have a systematic bias in shift or scaling by design) but can mask systematic errors that may arise when a model is used in production.

Absolute error

1. **Absolute error**

```
(sum(abs(d$prediction-d$y)))
```

2. **Mean absolute error**

```
(sum(abs(d$prediction-d$y))/length(d$y))
```

3. **Relative absolute error**

```
(sum(abs(d$prediction-d$y))/sum(abs(d$y)))
```

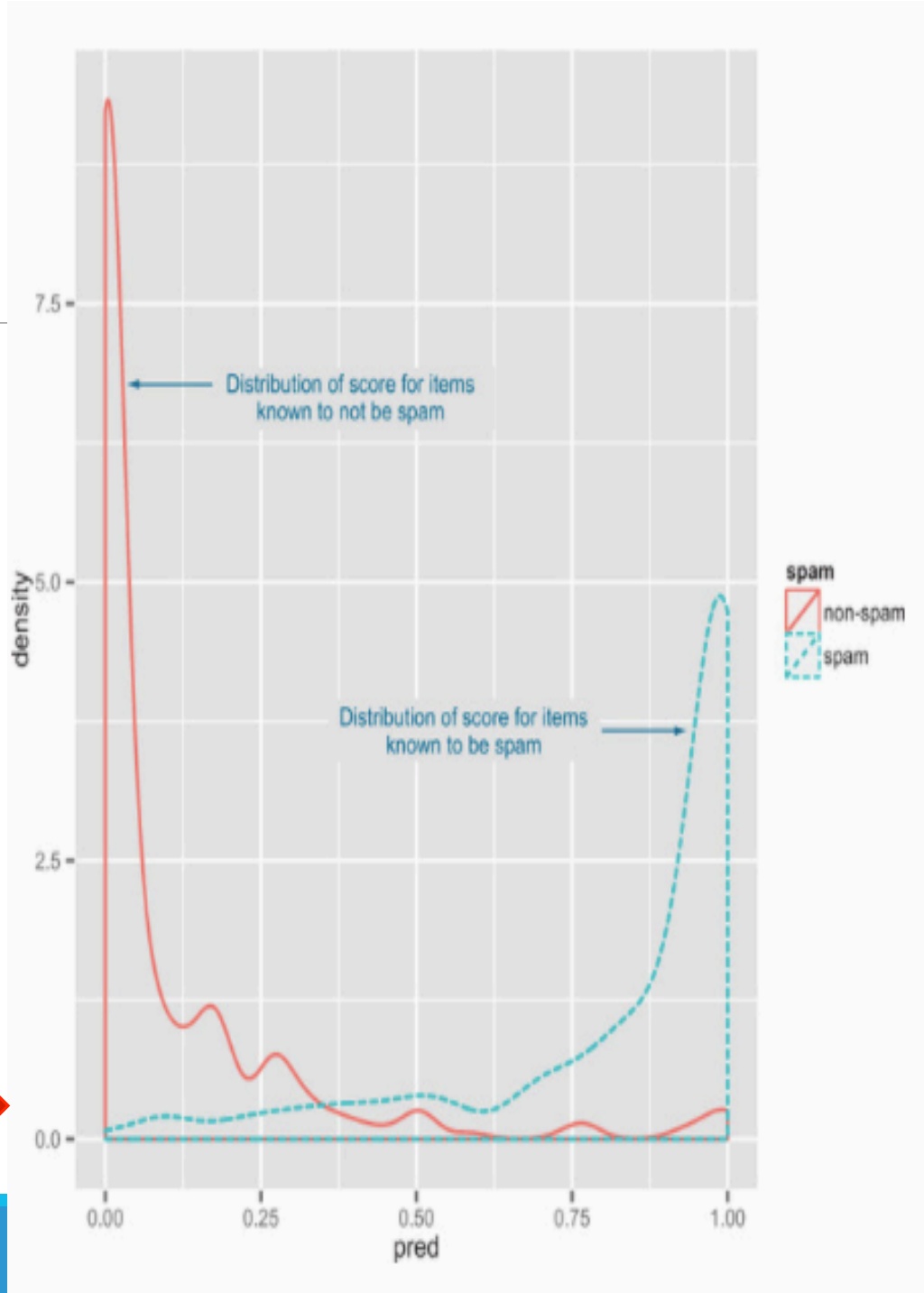
Absolute error

- Absolute error measures tend not to get aggregates right” or roll up reasonably” as most of the squared errors do.
- It is often an unstated project need that various totals or roll-ups of the predicted amounts be close to the roll-ups of the unknown values to be predicted.

Evaluating probability models

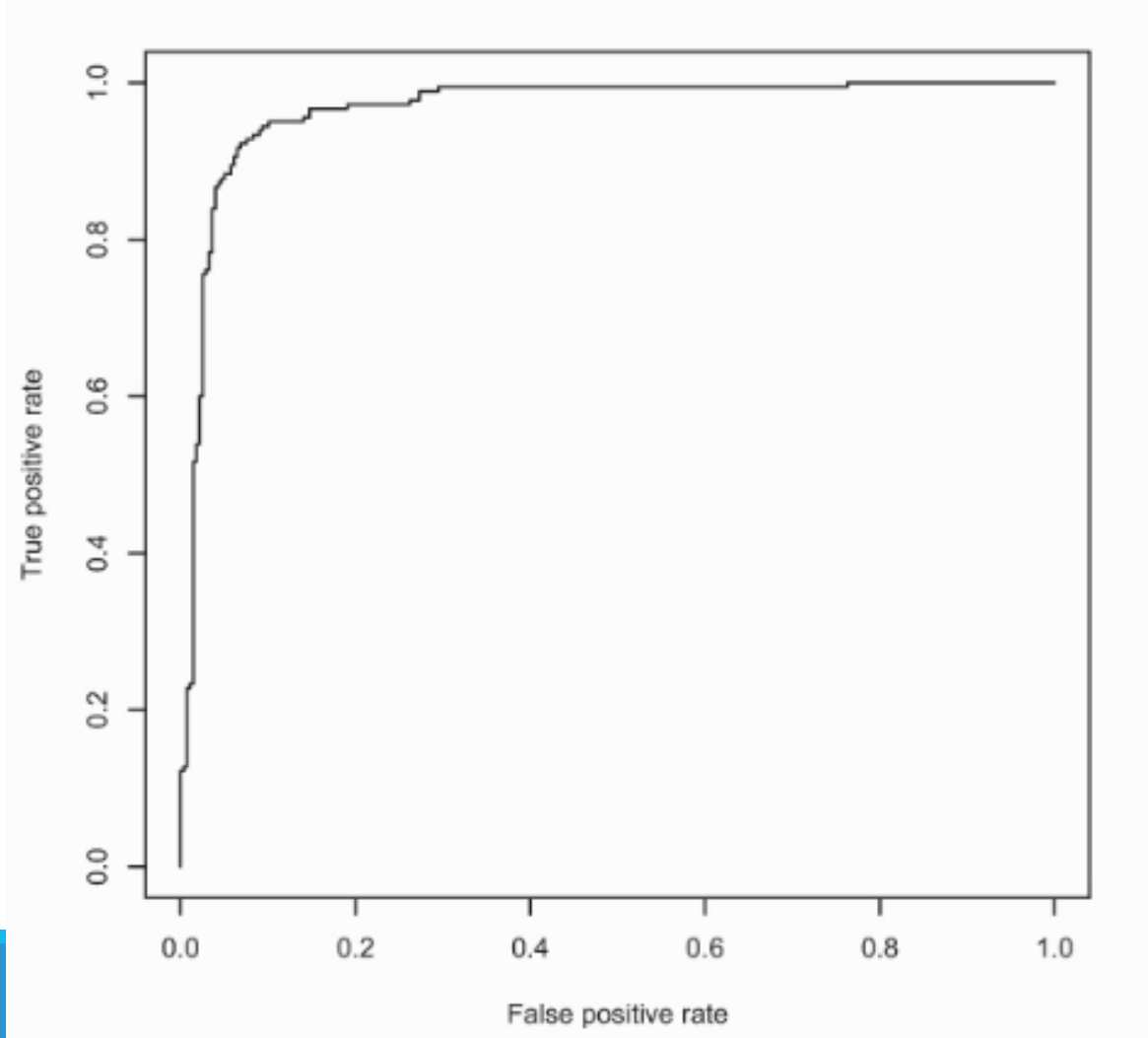
- Probability models are useful for both classification and scoring tasks
- Models that both decide if an item is in a given class and return an estimated probability (or confidence) of the item being in the class.

When thinking about probability models, it's useful to construct a double density plot:
Distribution of score broken up by known classes



The Receiver Operating Characteristic curve

- ROC curve is a popular alternative to the double density plot. For each different classifier we'd get by picking a different score threshold between positive and negative determination, we plot both the true positive rate and the false positive rate.



Log likelihood

- An important evaluation of an estimated probability is the log likelihood.
- The log likelihood is the logarithm of the product of the probability the model assigned to each example.
- If the model is a good explanation, then the data should look likely (not implausible) under the model.

Deviance

- Deviance is defined as `-2*(logLikelihood-S)`, where *S* is a technical constant called the log likelihood of the saturated model."
- The lower the residual deviance, the better the model.
- In most cases, the saturated model is a perfect model that returns probability 1 for items in the class and probability 0 for items not in the class (so S=0).

AIC

- Akaike information criterion (AIC). This is equivalent to `deviance + 2*numberOfParameters` used in the model used to make the prediction.
- AIC is deviance penalized for model complexity.
- A nice trick is to do as the Bayesians do: use Bayesian information criterion (BIC) (instead of AIC) where an empirical estimate of the model complexity (such as `2*2^entropy,` instead of `2*numberOfParameters`)
- AIC is useful for comparing models with different measures of complexity and variables with differing number of levels.

Entropy

- Entropy is a fairly technical measure of information or surprise, and is measured in a unit called bits. If p is a vector containing the probability of each possible outcome, then the entropy of the outcomes is calculated as sum(-p*log(p,2)) (with the convention that 0*log(0) = 0).
- ***conditional entropy*** is a measure that gives an indication of how good the prediction is on different categories, tempered by how often it predicts different categories.

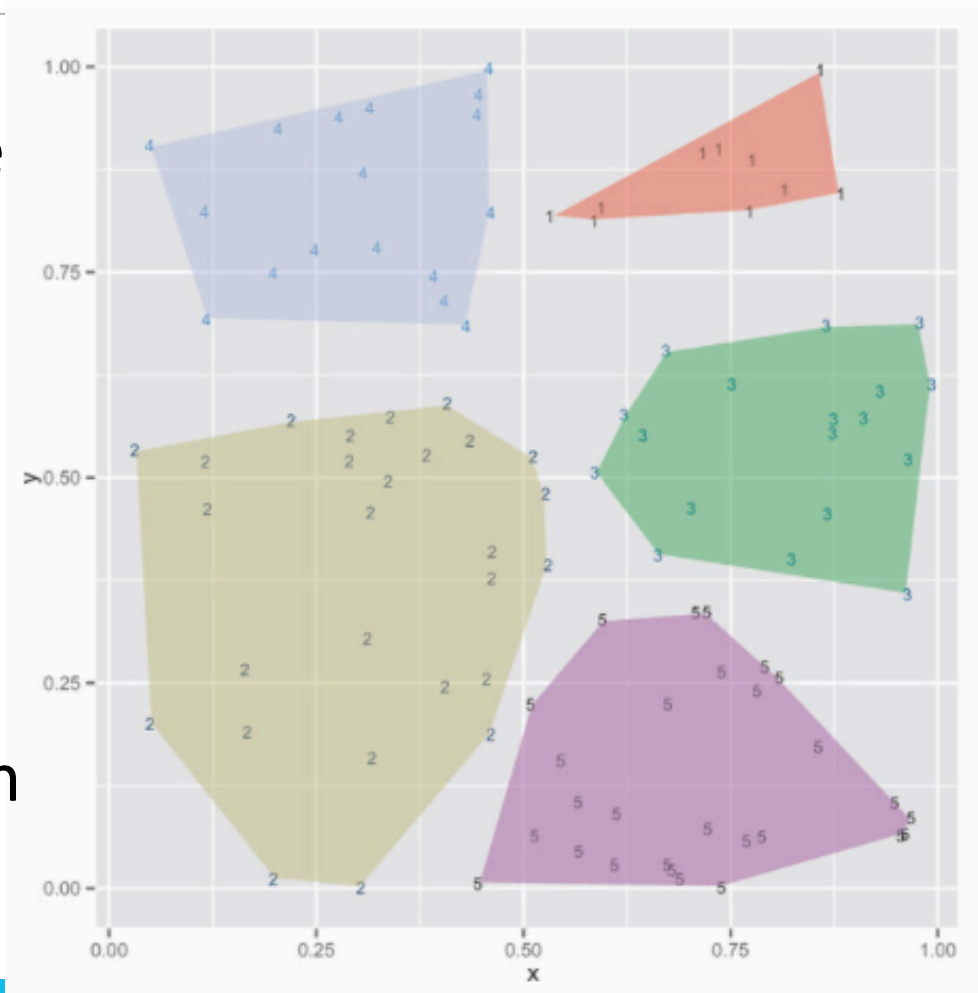
Evaluating ranking models

- Ranking models are models that, given a set of examples, sort the rows or (equivalently) assign ranks to the rows.
- Ranking models are often trained by converting groups of examples into many pair-wise decisions (statements like a is before
- Standard measures of a ranking model are:

1. Spearman's rank correlation coefficient (treating assigned rank as a numeric score) and;
2. The data mining concept of lift (treating ranking as sorting.

Evaluating clustering models

- Clustering models are hard to evaluate because they're unsupervised: the clusters that items are assigned to are generated by the modeling procedure, not supplied in a series of annotated examples. Evaluation is largely checking observable summaries about the clustering.
- The first qualitative metrics are: 1-how many clusters you have (sometimes chosen by the user, or algorithm) and 2- the number of items in each cluster.



Intra-cluster distances versus cross-cluster distances

- A desirable feature in clusters is for them to be compact in whatever distance scheme you used to define them

comparing the typical distance between two items in the same:

- A desirable feature in clusters is for them to be compact in whatever distance scheme you used to define them.
- The traditional measure of this is comparing the typical distance between two items in the same cluster to the typical distance between two items from different clusters.

Treating clusters as classifications or scores

- Distance metrics are good for checking the performance of clustering algorithms
- they don’t always translate to business needs
- For each cluster label, generate an outcome assigned to the cluster
- Then use either the classifier or scoring model evaluation ideas to evaluate the value of the clustering

Validating models

- Will it show similar quality on new data in production?
- ***model validation***: testing of a model on new data (or a simulation of new data from our test set.
- Identifying common model problems

Identifying common model problems

Problem	Description
Bias	Systematic error in the model, such as always underpredicting.
Variance	Undesirable (but non-systematic) distance between predictions and actual values. Often due to oversensitivity of the model training procedure to small variations in the distribution of training data.
Overfit	Features of the model that arise from relations that are in the training data, but not representative of the general population. Overfit can usually be reduced by acquiring more training data and by techniques like regularization and bagging.
Non-significance	A model that appears to show an important relation when in fact the relation may not hold in the general population, or equally good predictions can be made without the relation.

Overfitting

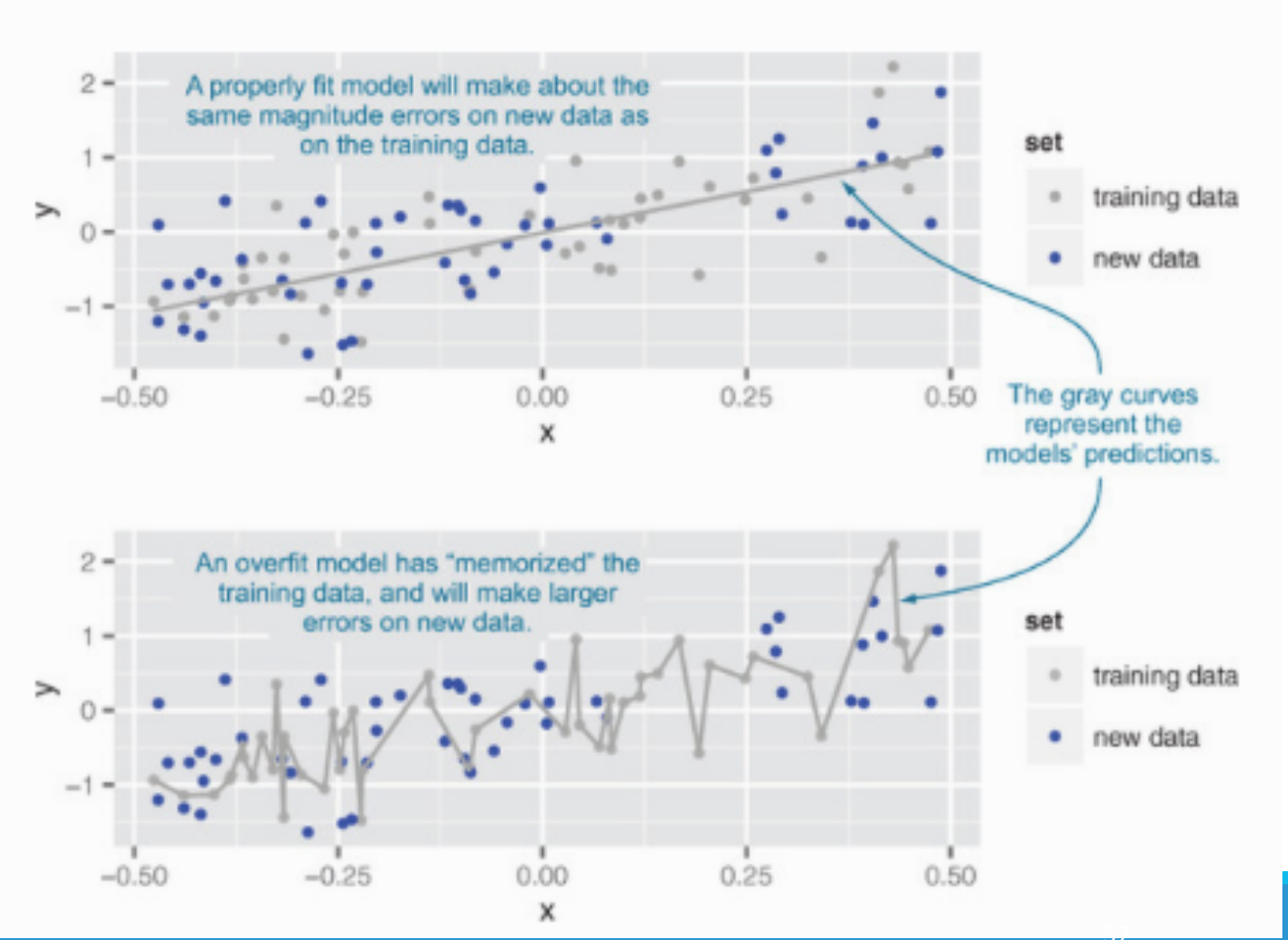
- A lot of modeling problems are related to overfitting. Looking for signs of overfit is a good first step in diagnosing models.
- An overfit model looks great on the training data and performs poorly on new data.
- A model's prediction error on the data that it trained from is called *training error.*
- A model's prediction error on new data is called *generalization error*.
- Usually, training error will be smaller than generalization error.
- Ideally, though, the two error rates should be close. If generalization error is large, then your model has probably overfit—it's memorized the training data instead of discovering generalizable rules or patterns.

A notional illustration of overfitting

An overly complicated and overfit model is bad for at least two reasons:

1- an overfit model may be much more complicated than anything useful.

2- overfit models tend to be less accurate in production than during training, which is embarrassing.



Quantifying model soundness

- Evaluating a model only on the data used to construct it is favorably biased:
- a single evaluation of model performance gives only a *point estimate* of performance.

Frequentist and Bayesian inference

- In frequentist inference, we assume there is a single unknown *fixed* quantity
- The frequentist inference for a given dataset is a *point estimate* that varies as different possible datasets are proposed (yielding a distribution of estimates).
- In Bayesian inference, we assume that the unknown quantity to be estimated has many plausible values modeled by what's called a prior distribution.
- Bayesian inference is then using data (that is considered as unchanging) to build a tighter posterior distribution for the unknown quantity.
- In practice, choosing your inference framework isn't a matter of taste, but a direct consequence of what sort of business question you're trying to answer. If you're worried about the sensitivity of your result to possible variation in the unknown quantity to be modeled, you should work in the Bayesian framework

Ensuring model quality

- Always share how sensitive your conclusions are to variations in your data and procedures.
- You should never just show a model and its quality statistics.
- You should also show the likely distribution of the statistics under variations in your modeling procedure or your data.

Testing on held-out data

- The data used to build a model is not the best data for testing the model's performance. This is because there's an upward measurement bias in this data.
- Split your available data into test and training. Perform all of your clever work on the training data alone, and delay measuring your performance with respect to your test data until as late as possible in the project.

K-fold cross-validation

- In practice we want both an **unbiased estimate** of our model's future performance on new data (simulated by test data) and **an estimate** of the distribution of this estimate under typical variations in data and training procedures.
- A good method to perform these estimates is k-fold cross-validation and the related ideas of empirical resampling and bootstrapping.
- k-fold cross-validation is to repeat the construction of the model on different subsets of the available training data and then evaluate the model only on data not seen during construction.

Significance testing

- Statisticians have a powerful idea related to cross-validation called significance testing.
- Significance also goes under the name of p-value and you will actually be asked, “What is your p-value?” when presenting.
- When you evaluate your model over a test set of houses, you will (hopefully) see that `D = (err.null - err.model) > 0` (your model is more accurate)
- Always think about p-values as estimates of how often you’d find a relation (between the model and the data) when there actually is none. This is why low p-values are good, as they’re estimates of the probabilities of undetected disastrous situations

Confidence intervals

- A 95% confidence interval is an interval from an estimation *procedure* such that the procedure is thought to have a 95% (or better) chance of catching the true unknown value to be estimated in an interval.
- It is not the case that there is a 95% chance that the unknown true value is actually in the interval at hand (thought it's often misstated as such).

Using statistical terminology

- The field of statistics has spent the most time formally studying the issues of model correctness and model soundness
- Because of their priority, statisticians often insist that the checking of model performance and soundness be solely described in traditional statistical terms.
- It's not always practical to allow the dictates of a single field to completely style a cross-disciplinary conversation.

Key Takeaways

- Always first explore your data, but don't start modeling before designing some measurable goals.
- Divide you model testing into establishing the model's effect (performance on various metrics) and soundness (likelihood of being a correct model versus arising from overfitting).
- Keep a portion of your data out of your modeling work for final testing. You may also want to subdivide your training data into training and calibration and to estimate best values for various modeling parameters.
- Keep many different model metrics in mind, and for a given project try to pick the metrics that best model your intended business goal.