Machine Learning with Python

Session 1: Introduction to Machine Learning, Data Exploration and Data Visualization using Python

Arghya Ray

What is machine learning?

Machine learning is the science (and art) of programming computers so that they can learn from the data.

"Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed."

- Arthur Samuel, 1959

"A computer program is said to learn from experience **E** with respect to some task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improves with experience **E**." - Tom Mitchell, 1997

Why use Machine Learning?

Machine Learning is great for:

- Problems for which existing solutions require a lot of hand tuning or long lists of rules: *One Machine Learning algorithm can often simplify code and perform better.*
- Complex problems for which there is no good solution at all using a traditional approach: *the best Machine Learning techniques can find a solution.*
- Fluctuating environments: *a Machine Learning system can adapt to new data.*
- Getting insights about complex problems and large amount of data.

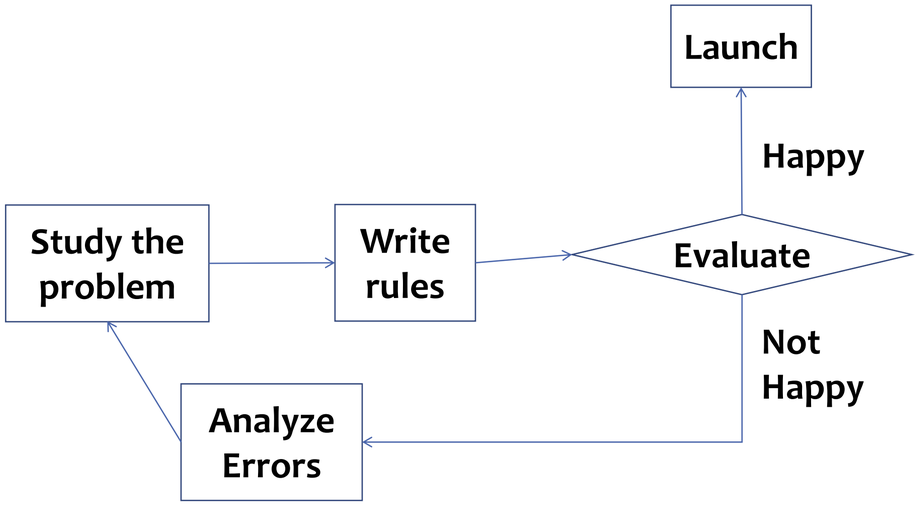


Figure 1. The traditional approach

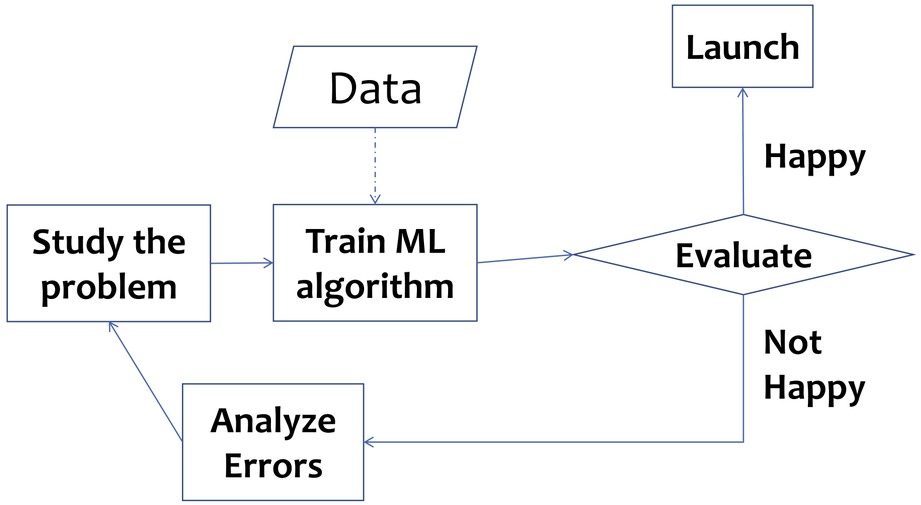


Figure 2. The Machine learning approach

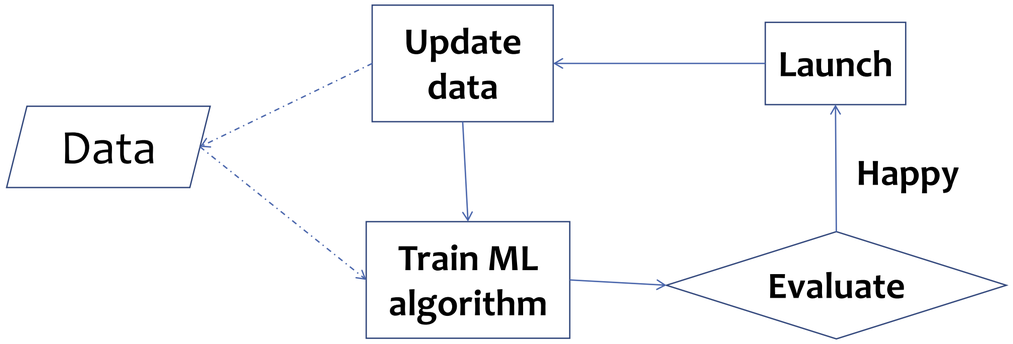


Figure 2. The machine learnng approach

Data Mining is a collection of techniques for efficient discovery of previously unknown, valid, novel, useful and understandable patterns in large databases. The patterns must be actionable so that they may be used in an enterprise's decision making.

Why do we need data-mining now?

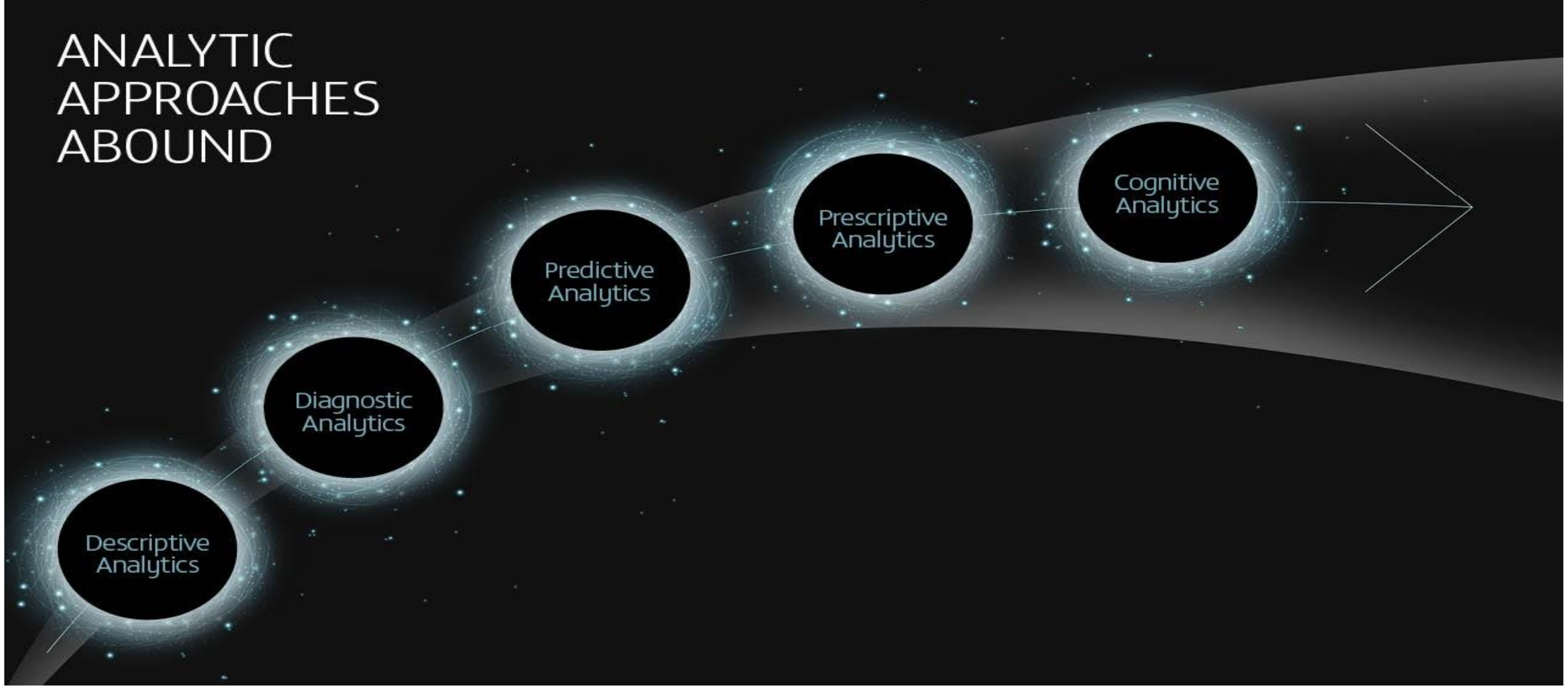
- Growth in data
- Decline in cost of processing
- Growth in data storage capacity
- Competitive environment
- Availability of various data-mining softwares

Data-Mining Applications:

- Prediction and Description
- Relationship Marketing
- Customer Profiling and Customer Segmentation
- Outlier Identification and Fraud Detection

Domains where data-mining is used:

- Astronomy
- Banking and Finance
- Business
- Crime Prevention
- Education
- Government
- Health-care
- Manufacturing
- Telecommunications
- Transportation



Data Analytics has evolved over the years from **Descriptive** (*what has happened*) to **Diagnostic** (*why did it happen*) to **Predictive** (*what could happen*) to **Prescriptive** (*what action could be taken*).

The next big paradigm shift will be towards **Cognitive Analytics** which will exploit the massive advances in High Performance Computing by combining advanced Artificial Intelligence and Machine Learning techniques with data analytics approaches.

Descriptive Analytics	Descriptive analytics is the interpretation of historical data to better understand changes that have occurred in a business. E.g., Year over year pricing changes, month over month sales growth, etc.
Predictive Analytics	"The purpose of predictive analytics is not to tell you what will happen in future. It cannot do that. In fact no analytics can do that. Predictive Analytics can only forecast what might happen in future, because all predictive analytics are probabilistic in nature," – Dr. Michael Wu, Lithium Technologies. The three keystones of predictive analytics are: (a) decision analysis and optimization; (b) transactional profiling; (c) predictive modeling. Predictive Analytics exploits patterns in transactional and historical data to identify risks and opportunities,
Prescriptive Analytics	Prescriptive Analytics is an emerging discipline and represents a more advanced use of predictive analytics. Prescriptive analytics goes beyond simply predicting options in the predictive model and actually suggests a range of prescribed actions and the potential outcomes for each action. Dr. Wu said that "Since a prescriptive model is able to predict the possible consequences based on different choices of action, it can also recommend the best course of action for any pre-specified outcome." E.g., 1. Google' self driving car 2. In the energy sector, gas producers and pipeline companies use prescriptive analytics to identify factors affecting the price of oil and gas.

Business Intelligence and Business Analytics:

In Industry, business intelligence is using BI tools to get some information. The data is used form data warehouse to get some answers from certain queries or generate reports. BI uses historical data.

E.g.: As far as banking loans are concerned, how does a customer behave – before marriage and after marriage.

What is the purchase pattern of customers on weekends in various cities.

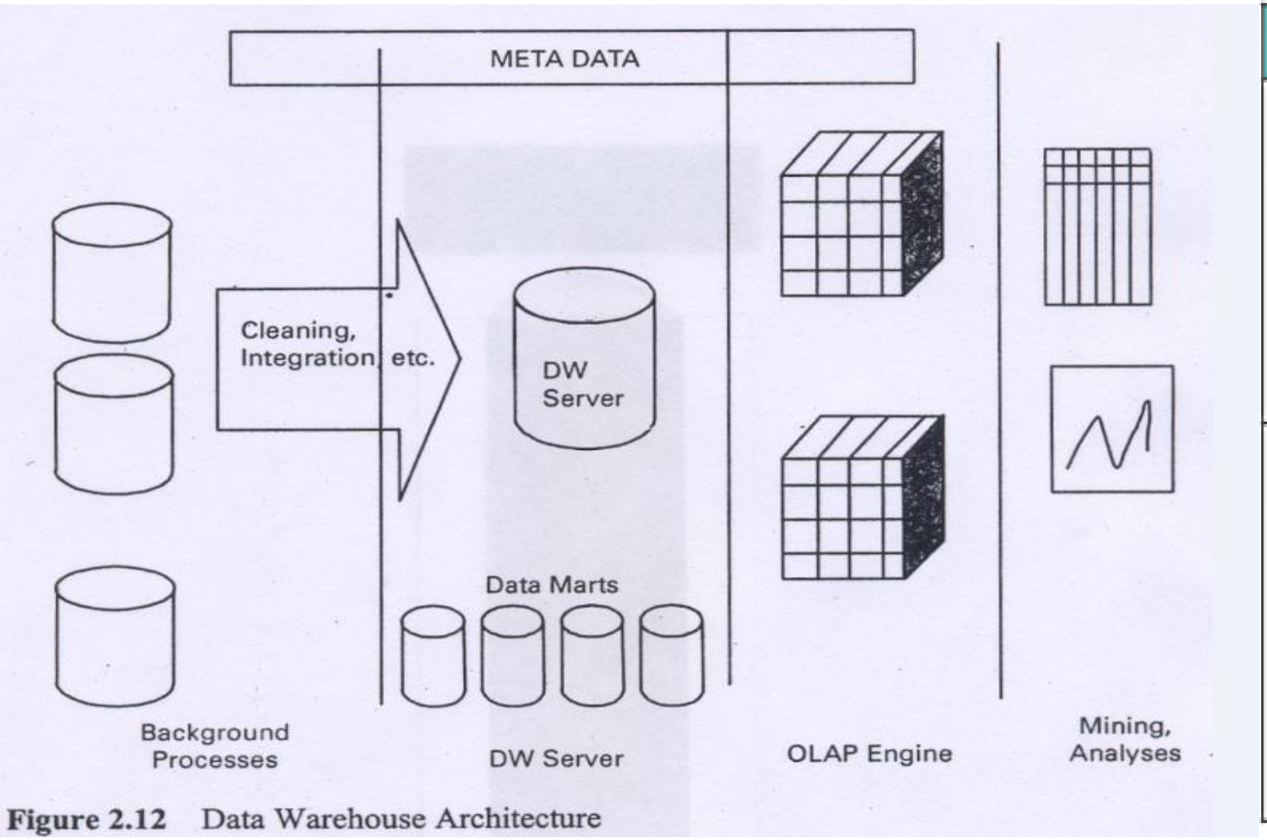


Figure 2.12 Data Warehouse Architecture

BI vs BA	Business Intelligence	Business Analytics
Answers the questions:	What happened? When? Who? How many?	Why did it happen? Will it happen again? What will happen if we change *x*? What else does the data tell us that never thought to ask?
Includes:	Reporting (KPIs, metrics) Automated Monitoring/Alerting (thresholds) Dashboards Scorecards OLAP (Cubes, Slice & Dice, Drilling) Ad hoc query	Statistical/Quantitative Analysis Data Mining Predictive Modeling Multivariate Testing

Difference between predictive analytics and business intelligence:

Business Intelligence answers the question, "From what ZIP code does my most valuable customers come?"

Predictive analytics however answers, "How much revenue can I expect from customers in a particular ZIP code?"

Types of Machine Learning Systems

Broad Categories of Machine Learning Systems:

- ***Whether or not they are trained with human supervision***
  - Supervised
  - Unsupervised
  - Semi-supervised
  - Reinforcement
- ***Whether or not they can learn incrementally on the fly***
  - Online Learning
  - Batch Learning
- ***Whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model***
  - Instance based learning
  - Model based learning

Machine Learning systems can be classified according to the amount and type of supervision they get during training.

Supervised Learning:

Goal: Predict a single "target" or "outcome" variable.

- Has definite outcomes or goals.
- Predict answers for new unknown values

Training data, where target value is known

Score to data where value is not known

Methods: Classification and Prediction

Unsupervised Learning:

Goal: Segment data into meaningful segments; detect patterns.

- No target (outcome) variable to predict or classify
- It makes sense of data from observations.

Methods: Association rules, data reduction & exploration, visualization, Anomaly detection

Semi-Supervised Learning:

Some algorithms can deal with partially labeled training data, usually a lot of unlabeled data and a little bit of labeled data.

E.g.: Some photo-hosting services, such as Google Photos, are good examples of this.

Reinforcement Learning:

The learning system, called an agent in this context, can observe the environment, select and perform actions, and get rewards in return (or penalties in the form of negative rewards). It must then learn by itself what is the best strategy, called a policy, to get the most reward over time.

Machine Learning systems can be classified based on whether or not the system can learn incrementally from a stream of incoming data.

Batch Learning/Offline Learning:

- In batch learning, the system is incapable of learning incrementally: it must be trained using all the available data. This will generally take a lot of time and computing resources, so it is typically done offline.
- First the system is trained, and then it is launched into production and runs without learning anymore; it just applies what it has learned.
- If you have a lot of data and you automate your system to train from scratch every day, it will end up costing you a lot of money. If the amount of data is huge, it may even be impossible to use a batch learning algorithm.

Incremental Learning/Online Learning:

- In online learning, you train the system incrementally by feeding it data instances sequentially, either individually or by small groups called mini-batches.
- One important parameter of online learning systems is how fast they should adapt to changing data: this is called the ***learning rate***.
- If you set a high learning rate, then your system will rapidly adapt to new data, but it will also tend to quickly forget the old data (you don't want a spam filter to flag only the latest kinds of spam it was shown).
- Conversely, if you set a low learning rate, the system will have more inertia; that is, it will learn more slowly, but it will also be less sensitive to noise in the new data or to sequences of non-representative data points.
- If bad data is fed into the system, the system's performance will gradually decline.

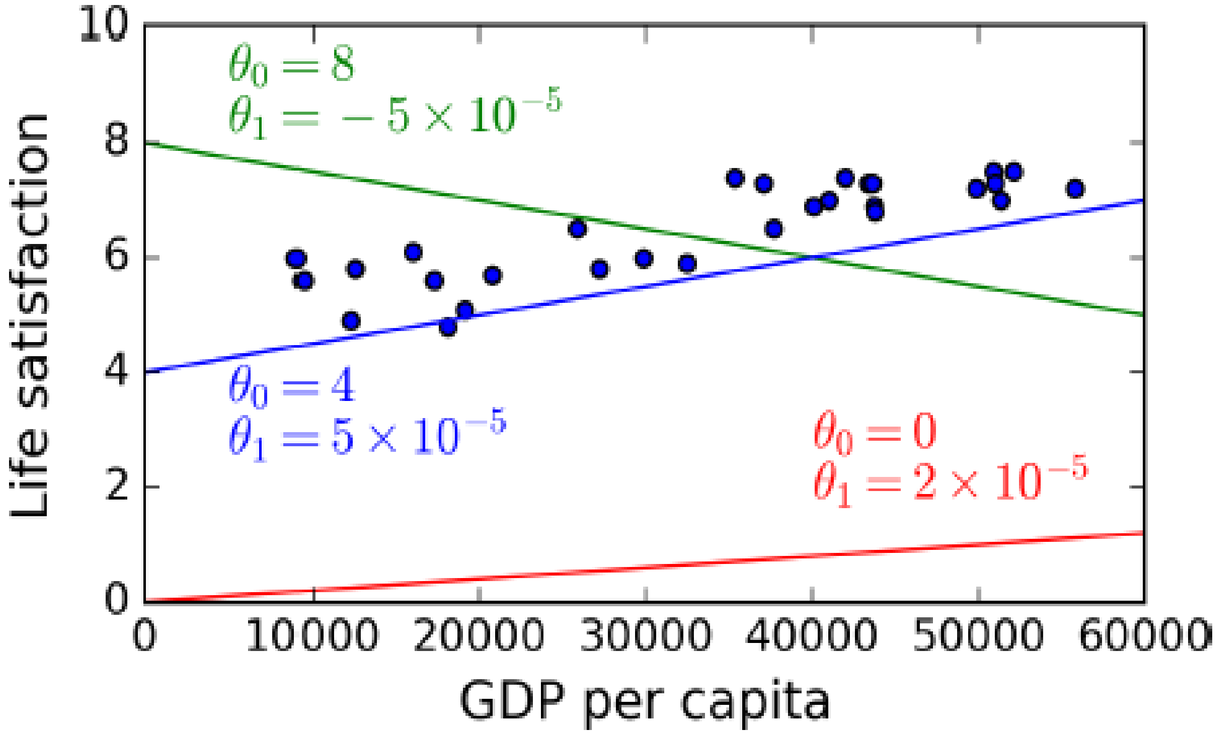
One more way to categorize Machine Learning systems is by how they generalize.
Having a good performance measure on the training data is good, but insufficient; the true goal is to perform well on new instances.

Instance-based Learning:

- Possibly the most trivial form of learning is simply to learn by heart.
- If you were to create a spam filter this way, it would just flag all emails that are identical to emails that have already been flagged by users—not the worst solution, but certainly not the best. Instead of just flagging emails that are identical to known spam emails, your spam filter could be programmed to also flag emails that are very similar to known spam emails. This requires a measure of similarity between two emails.

Model-based Learning:

- Another way to generalize from a set of examples is to build a model of these examples, then use that model to make predictions..



Supervised vs Unsupervised Learning

- Deciding whether to issue a loan to an applicant based on demographic and financial data (with reference to a database of similar data on prior customers).
- In an online bookstore, making recommendations to customers concerning additional items to buy based on the buying pattern in prior transactions.
- Identifying a network data packet as dangerous (virus, hacker attack) based on comparison to other packets whose threat status is known.
- Identifying segments of similar customers.
- Predicting whether a company will go bankrupt or not based on comparing its financial data to those of similar bankrupt and non-bankrupt firms.
- Automated sorting of mail by zip-code scanning.
- Estimating the repair time required for an aircraft based on a trouble ticket.
- Printing of custom discount coupons at the conclusion of a grocery store checkout based on what you just bought and what others have bought previously.

Main Challenges of Machine Learning:

- Insufficient quantity of training data
- Non-representative training data → sampling noise. Sampling bias
- Poor quality data → standard format, 5,10,15,30, 225,
- Irrelevant features → P, Tw, GN, ExS (Feature Selection, Feature extraction)
- Overfitting the training data (it means that the model performs well on the training data, but it does not generalize well)
    - Overfitting happens when the model is too complex relative to the amount and noisiness of the training data
    - Constraining a model to make it simpler and reduce the risk of overfitting is called **regularization**.
    - Possible solutions:
        - To simplify the model by selecting one with fewer parameters, by reducing the number of attributes in the training data or by constraining the models.
        - To gather more training data
        - To reduce the noise in the training data (e.g., fix data errors and remove outliers)
- Underfitting the training data (it occurs when your model is too simple to learn the underlying structure of the data)
    - Possible solutions:
        - Selecting a more powerful model, with more parameters
        - Feeding better features to the learning algorithms (feature engineering)
        - Reducing the constraints on the model (e.g., reducing the regularization hyperparameters)

Testing and Validating

The content of the slides are prepared from different textbooks.

References:

- Links:
  - https://www.sas.com/en_in/insights/big-data/what-is-big-data.html
  - https://www.oracle.com/big-data/what-is-big-data/
  - https://www.w3schools.com/python/python_variables_multiple.asp

- Predictive Analytics for Dummies, By Anasse Bari, Mohamed Chaouchi, & Tommy Jung, Copyright 2016, John Wiley & Sons, Inc.
- Introduction to Data Mining with Case Studies, By G.K. Gupta. Copyright 2014 by PHI Learning Private Limited.


Thank you..

Machine Learning with Python

Session 2: End-to-end Machine Learning Project

Arghya Ray

Main steps you need to go through:

1. Look at the big picture (what is the objective, frame the problem, what type of algorithm to use, performance measures)
2. Get the data (get a quick view of data using head(), info(), value_counts(), describe(),etc.)
3. Discover and visualize the data to gain insights (generalization error→ data snooping bias; Finding correlations)
4. Prepare the data for Machine Learning Algorithms (data cleansing, handling text and categorical attributes, custom transformers, feature scaling→Min-max scaling, standardization)
5. Select a model and train it (Split into training and testing sets, training and evaluating, Better evaluation using cross-validation)
6. Fine tune your model (Grid-search, Randomized search, Ensemble methods)
7. Present your solution (analyze the best models and their errors)
8. Launch, Monitor and Maintain your system

Data Collection and Pre-processing:

- Improving the quality of data in databases for use in data-mining is a challenging task. The presence of incorrect and inconsistent data can significantly impact the result of data mining analysis and therefore potential benefits of using data-mining may not be achieved.
- Usually data required for data mining tasks needs to be extracted from a number of databases, integrated and perhaps cleansed and transformed. This process is called ***ETL (Extraction, Transformation and Loading)***.
- **Data Cleansing** is a process used to determine inaccurate, incomplete or unreasonable data items of a dataset and then improving the data quality through corrections of the detected errors and omissions.

- **Sources of errors in the data:**
  - **Instance Identity Errors**: Same individual may be represented slightly differently in different source systems.
  - **Data Errors**: Deals with missing attribute values, duplicate records, wrong aggregations, non-unique identifiers, inconsistent use of nulls spaces and empty spaces, coding mismatch across databases, inappropriate use of address lines, etc.
  - **Record Linkage Problem**: The problem of linking information from different databases that relates to the same customer or client.
  - **Semantic Integration Problem**: Deals with errors that arise during integration of information found in different sources.
  - **Data Integrity Problem**: Data integrity deals with issues like referential integrity, null values, domain of values, etc.
  - **Data Entry Errors**: Due to unmotivated data entry staff.
  - **Measurement Errors**: Errors creep in because of instrument malfunctioning, poor calibration, or poor design of s/w used in instrument.
  - **Filtering Errors**: Each step of filtering, smoothing, and summarization of data is prone to produce errors.

Detecting Outliers:

- An **outlier** is an observation that is "extreme", being distant from the rest of the data (definition of "distant" is deliberately vague)
- Different data mining software appear to include different criteria for identifying outliers.
- Outliers can have disproportionate influence on models. Detecting outliers is an important step in data pre-processing.
- Once detected, domain knowledge is required to determine if it is an error, or truly extreme.
- Even though it is often thought that outliers should be quickly eliminated, but outliers can contain useful information. Some cases:
  - In a dataset about number of visas or passports issued by different offices or branches in a country, an outlier may show that too many visas or passports were issued by one agency or branch.
  - In a dataset of expenditure incurred by each branch of a company, many overseas trips funded by one overseas branch of a MNC.
  - In a computer system that has software that monitors behaviour of its users, a user's behaviour may be found to be different than what is normally expected. This user may be flagged. Such an approach is used in what is called ***anomaly detection***.
  - Finding outliers is the purpose of the DM exercise (airport security screening). This is called "anomaly detection".
- Outliers may be of different types: **Univariate**, **Multivariate**, or **Time-series**.
- Some classify outliers are:
  - **Global Outliers**: When an outlier is significantly different from the rest of the data-points.
  - **Contextual Outliers**: When an outlier is significantly different from the rest of the data-points in the same context.
  - **Collective Outliers**: When a number of outliers are significantly different from the rest of the dataset.

Mining Outliers:

- **Mining Univariate Outliers:** A single dimension variable. Robust statistics to detect outliers: $(\mu - 3\sigma, \mu + 3\sigma)$
- **Mining Multivariate Outliers**: A multivariate dataset is a set of vectors, each data point being a vector. It is sometimes necessary to consider a number of attributes together like, population and population growth. Mean value and s.d. of the pair (x,y)
- **Distance based outliers**: In the discussion of outliers above, we have assumed that variables are normally distributed. In case the normality assumption is not true, a non-parametric model free approach is adopted that involves the pair wise distances.
- **Mining Time-series Outliers**: Time series data are mainly used for identifying seasonality, trend, etc. One technique is to use Mean absolute deviation (MAD).
- **Other Techniques**:
  - Some methods are based on classification methods- ***Supervised classification*** *and* ***Unsupervised Classification.***
  - Some outlier detection methods use **statistical tests** (Grubb's test) while others may use **distance-based approach** (Euclidian distance).
  - Outliers in some cases may be identified by examination of **unique rules** (Each value of the given attribute must be different from all other values of the attribute), **consecutive rules** (There can be no missing values between the lowest and highest values for the attribute and that all values must also be unique. E.g., as in check numbers), and **null rules** (Specifies the use of blanks, question marks, special characters or other strings that may indicate the null condition).
  - A common outlier detection method is the use of good data visualization software (histogram, box-plot, etc.).

Further Reading: https://towardsdatascience.com/assessing-the-quality-of-data-e5e996a1681b

Handling Missing Data:

- There can be a number of reasons for missing values including:
    - The particular data has no value associated with it.
    - The field was not applicable, the event did not happen, or the data was not available.
    - The person who entered the data did not know the right value or did not care if the value is filled in.
    - The value is to be provided by a later step of the process.
- Most algorithms will not process records with missing values. Default is to drop those records.
- **Solution 1: Omission**
    - If a small number of records have missing values, can omit them
    - If many records are missing values on a small set of variables, can drop those variables (or use proxies)
    - If many records have missing values, omission is not practical
- **Solution 2: Imputation**
    - Replace missing values with reasonable substitutes
    - Lets you keep the record and use the rest of its (non-missing) information

Normalizing (Standardizing) Data:

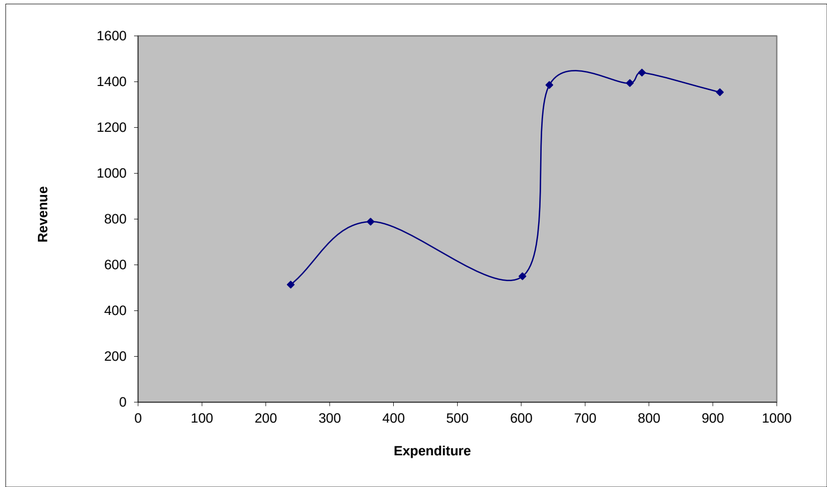
- Used in some techniques when variables with the largest scales would dominate and skew results
- Puts all variables on same scale
- Normalizing function: Subtract mean and divide by standard deviation (used in XLMiner)
- Alternative function: scale to 0-1 by subtracting minimum and dividing by the range

Rare event oversampling

- Often the event of interest is rare. Examples: response to mailing, fraud in taxes, etc.
- Sampling may yield too few "interesting" cases to effectively train a model
- Popular solution: oversample the rare cases to obtain a more balanced training set. Later, need to adjust results for oversampling.

The Problem of Over-fitting

- Statistical models can produce highly complex explanations of relationships between variables.
- The "fit" may be excellent. But when used with new data, models of great complexity do not do so well.
- Causes:
  - Too many predictors
  - A model with too many parameters
  - Trying many different models
- Consequence: Deployed model will not work as expected with completely new data.
- To handle the problem of over-fitting, we need to go for validation and testing.



Partitioning the Data:

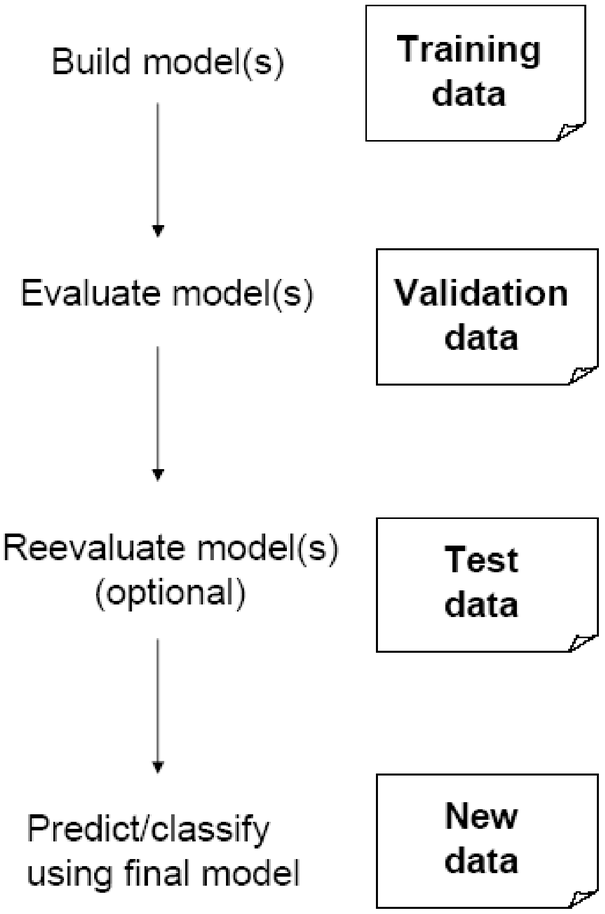
Problem: How well will our model perform with new data?

Solution: Separate data into two parts.

- Training partition to develop the model
- Validation partition to implement the model and evaluate its performance on "new" data.

Test Partition

- When a model is developed on training data, it can overfit the training data
  (hence need to assess on validation)
- Assessing multiple models on same validation data can overfit validation data.
- Some methods use the validation data to choose a parameter.
  This too can lead to overfitting the validation data .



Solution: final selected model is applied to a test partition to give unbiased estimate of its performance on new data

The content of the slides are prepared from different textbooks.

References:

- Links:
  - https://www.sas.com/en_in/insights/big-data/what-is-big-data.html
  - https://www.oracle.com/big-data/what-is-big-data/
  - https://www.w3schools.com/python/python_variables_multiple.asp

- Predictive Analytics for Dummies, By Anasse Bari, Mohamed Chaouchi, & Tommy Jung, Copyright 2016, John Wiley & Sons, Inc.
- Introduction to Data Mining with Case Studies, By G.K. Gupta. Copyright 2014 by PHI Learning Private Limited.


Thank you..

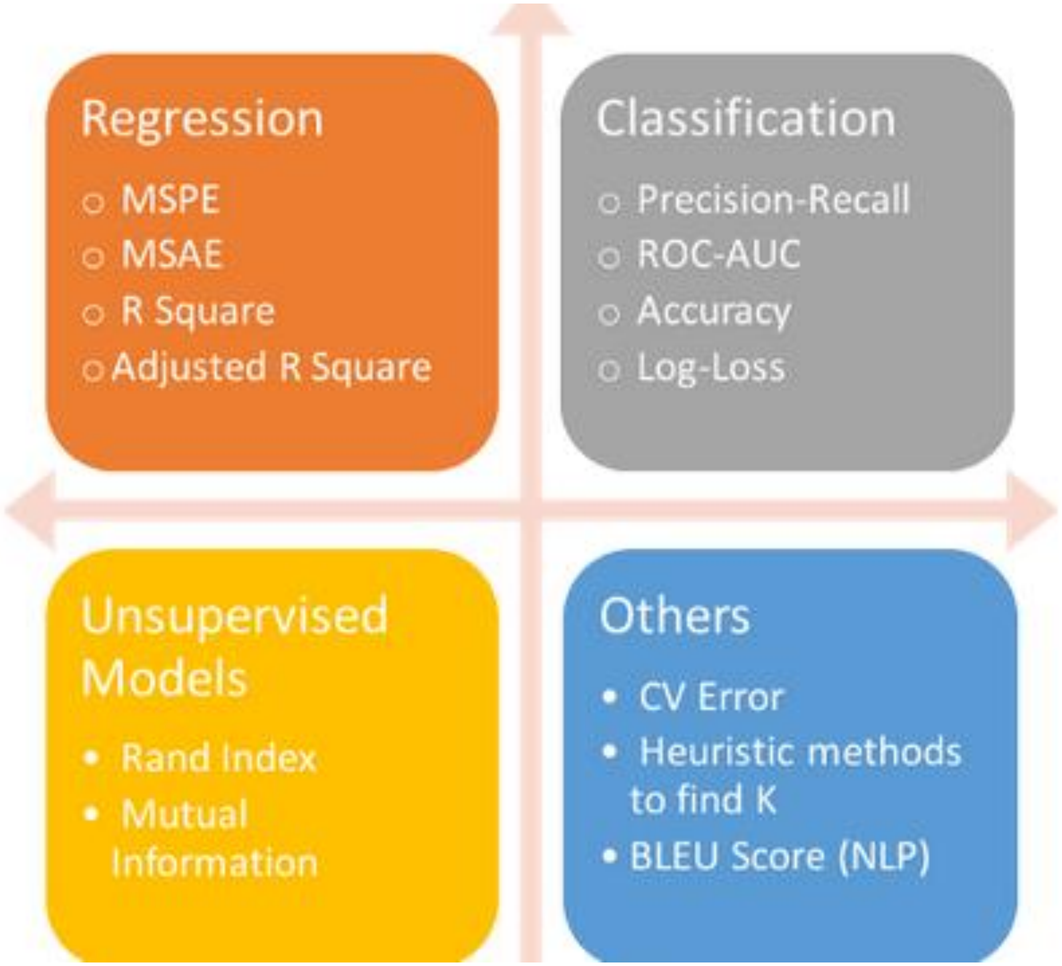
Machine Learning with Python

Measuring Performance of Classifiers

Arghya Ray

Why Evaluate?

- Multiple methods are available to classify or predict
- For each method, multiple choices are available for settings
- To choose best model, need to assess each model's performance



Reference: https://www.kaggle.com/usengecoder/performance-metrics-for-classification-problems

Accuracy Measures (Classification)

Misclassification error

- Error = classifying a record as belonging to one class when it belongs to another class.

- Error rate = percent of misclassified records out of the total records in the validation data

Naïve Rule

Naïve rule: classify all records as belonging to the most prevalent class

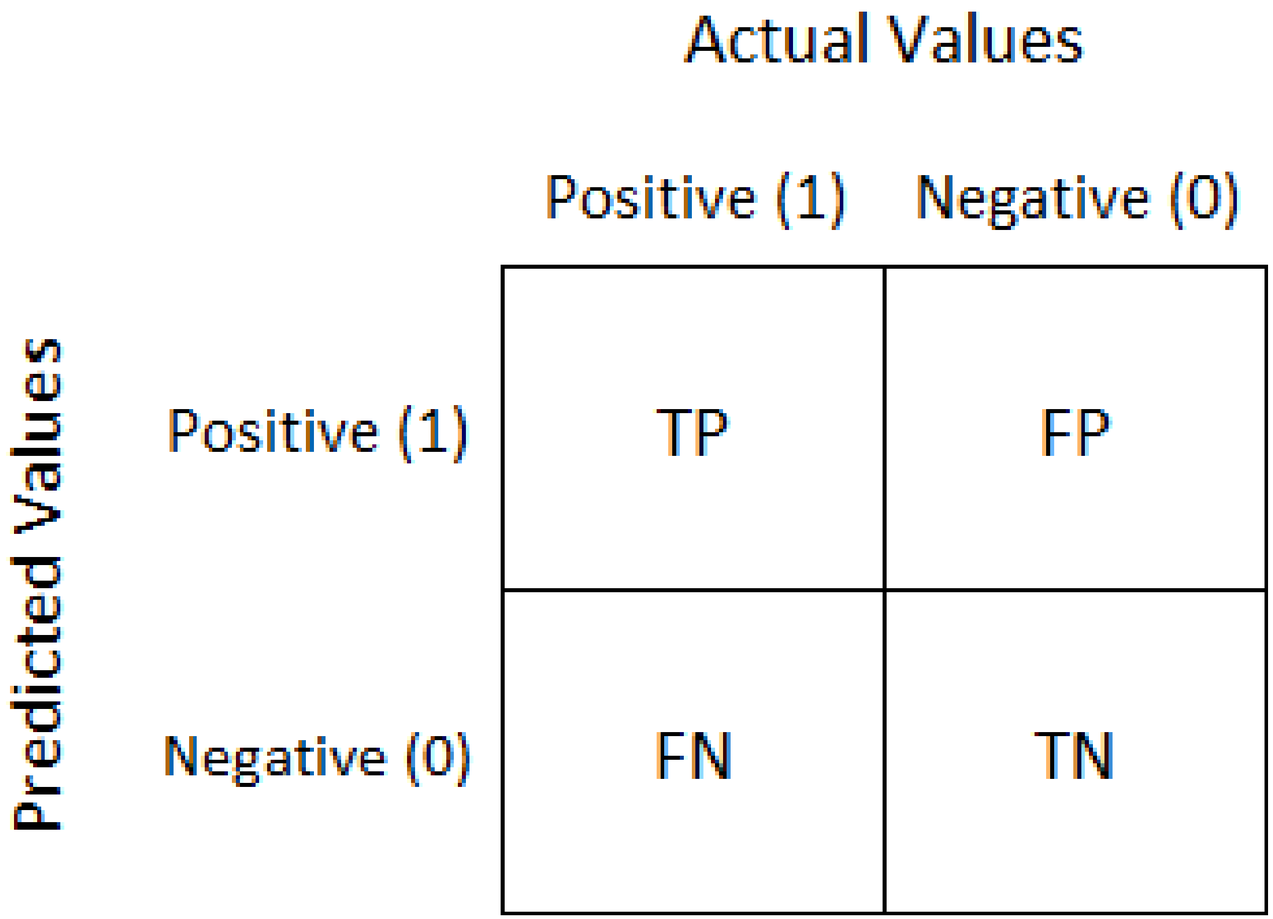
- Often used as benchmark: we hope to do better than that
- Exception: when goal is to identify high-value but rare outcomes, we may do well by doing worse than the naïve rule (see "lift" – later)

Separation of Records

“High separation of records” means that using predictor variables attains low error

“Low separation of records” means that using predictor variables does not improve much on naïve rule

Confusion Matrix



Total Value = (TP+FP+FN+TN)

Accuracy = (TP+TN)/Total values

1 – Accuracy = (FP+FN)/Total Values
=Error rate

Source: https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html

Confusion Matrix

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	201	85
0	25	2689

201 1's correctly classified as "1"

85 1's incorrectly classified as "0"

25 0's incorrectly classified as "1"

2689 0's correctly classified as "0"

Accuracy= (201+2689)/(201+85+25+2689)
= 0.96

Error rate = 1-0.96=0.04

Error Rate

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	201	85
0	25	2689

Overall error rate = (25+85)/3000 = 3.67%

Accuracy = 1 – err = (201+2689)/3000 = 96.33%

If multiple classes, error rate is:

(sum of misclassified records)/(total records)

Cutoff for classification

Most DM algorithms classify via a 2-step process:

For each record,

1. Compute **probability of belonging to class "1"**
2. Compare to cutoff value, and classify accordingly

- Default cutoff value is 0.50
  - If >= 0.50, classify as "1"
  - If < 0.50, classify as "0"
- Can use different cutoff values
- Typically, error rate is lowest for cutoff = 0.50

Cutoff Table

Actual Class	Prob. of "1"	Actual Class	Prob. of "1"
1	0.996	1	0.506
1	0.988	0	0.471
1	0.984	0	0.337
1	0.980	1	0.218
1	0.948	0	0.199
1	0.889	0	0.149
1	0.848	0	0.048
0	0.762	0	0.038
1	0.707	0	0.025
1	0.681	0	0.022
1	0.656	0	0.016
0	0.622	0	0.004

- If cutoff is 0.50: eleven records are actually in class “1”
- If cutoff is 0.80: seven records are actually in class “1”

Confusion Matrix for Different Cutoffs

Cut off Prob.Val. for Success (Updatable)	0.25

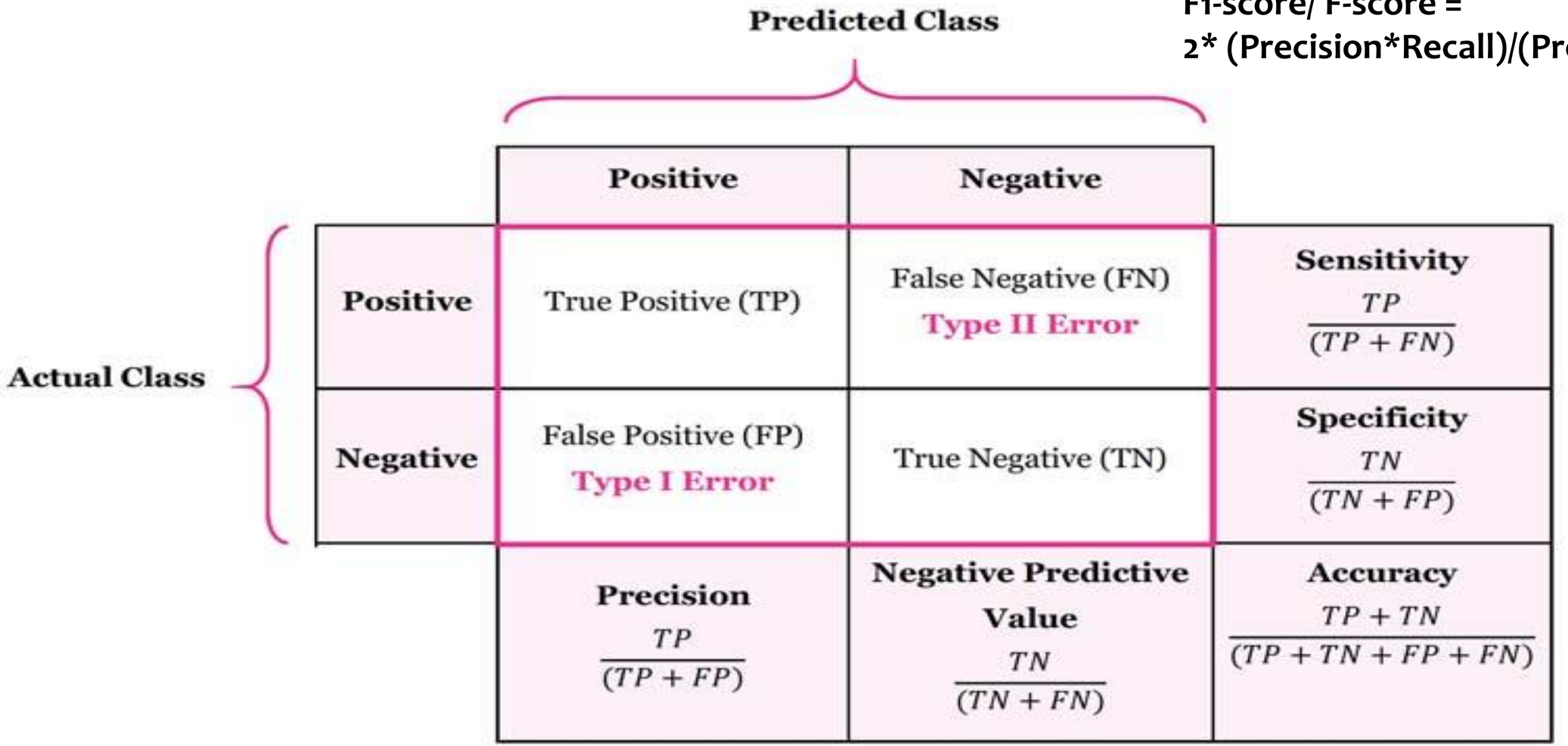
Classification Confusion Matrix		
	Predicted Class	
Actual Class	owner	non-owner
owner	11	1
non-owner	4	8

Cut off Prob.Val. for Success (Updatable)	0.75

Classification Confusion Matrix		
	Predicted Class	
Actual Class	owner	non-owner
owner	7	5
non-owner	1	11

Other performance measures.

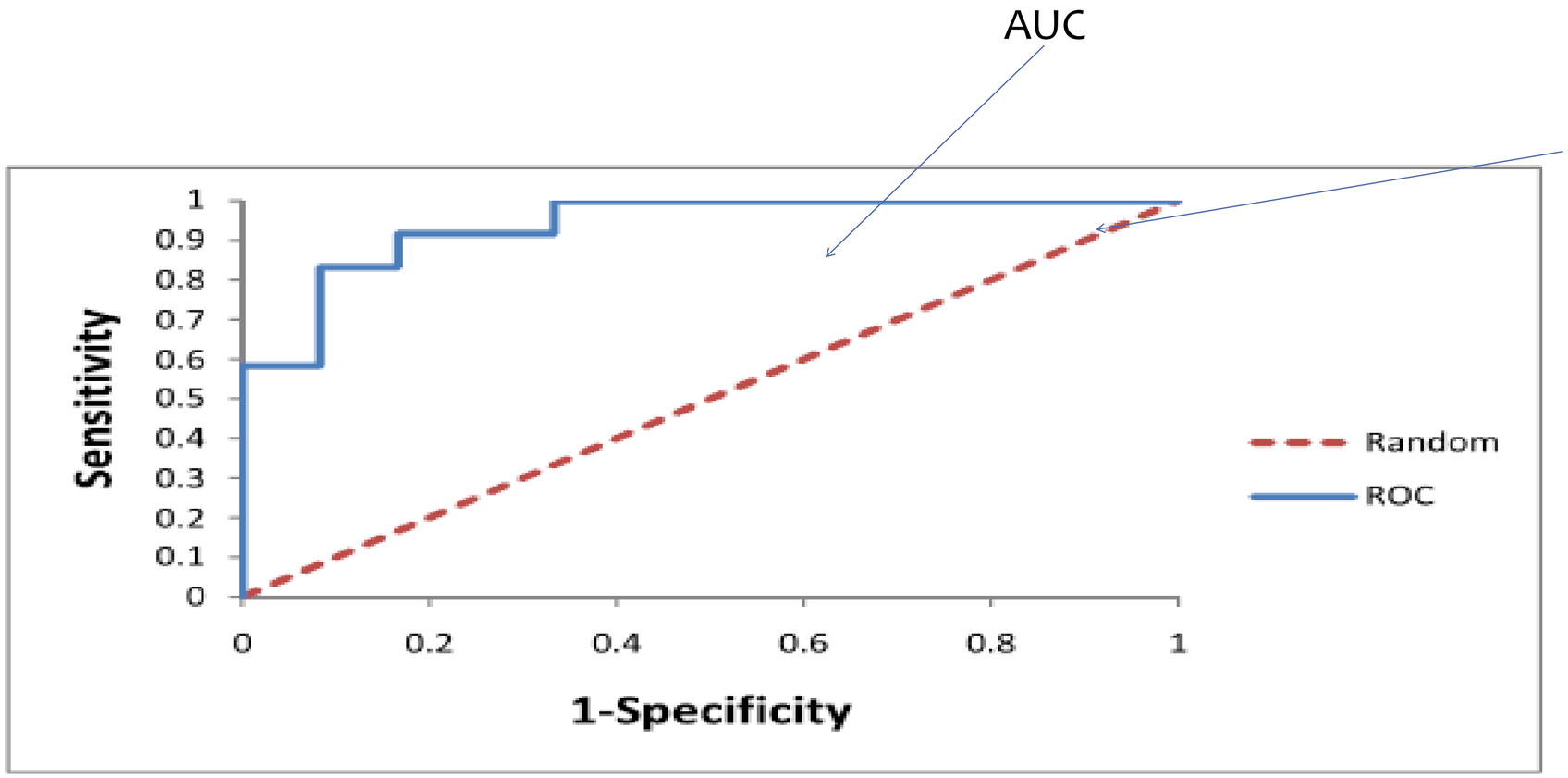
F1-score/ F-score =
2* (Precision*Recall)/(Precision+Recall)



Recall

Source: https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html

Receiver Operating Characteristic curve (ROC Curve)



Random line
Baseline model
Or
No Model
Or
Naïve Model

Along the random line,
Sensitivity = 1 - Specificity
i.e., the system does not know which class the customer belongs to.

A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

ROC Curve

Compare performance of DM model to “no model, pick randomly”

Measures ability of DM model to identify the important class, relative to its average prevalence

Charts give explicit assessment of results over a large number of cutoffs

Asymmetric Costs

Misclassification Costs May Differ

The cost of making a misclassification error may be higher for one class than the other(s)

Looked at another way, the benefit of making a correct classification may be higher for one class than the other(s)

Example – Response to Promotional Offer

Suppose we send an offer to 1000 people, with 1% average response rate

("1" = response, "0" = nonresponse)

- "Naïve rule" (classify everyone as "0") has error rate of 1% (seems good)

- Using DM we can correctly classify eight 1's as 1's

  It comes at the cost of misclassifying twenty 0's as 1's and two 0's as 1's.

The Confusion Matrix

	Predict as 1	Predict as 0
Actual 1	8	2
Actual 0	20	970

Error rate = (2+20) = 2.2% (higher than naïve rate)

Introducing Costs & Benefits

Suppose:

- Profit from a “1” is $10
- Cost of sending offer is $1

Then:

- Under naïve rule, all are classified as “0”, so no offers are sent: no cost, no profit
- Under DM predictions, 28 offers are sent.

  8 respond with profit of $10 each
  20 fail to respond, cost $1 each
  972 receive nothing (no cost, no profit)

- Net profit = $60

Profit Matrix

	Predict as 1	Predict as 0
Actual 1	$80	0
Actual 0	($20)	0

Generalize to Cost Ratio

Sometimes actual costs and benefits are hard to estimate

- Need to express everything in terms of costs (i.e., cost of misclassification per record)
- Goal is to minimize the average cost per record

A good practical substitute for individual costs is the **ratio** of misclassification costs (e,g,, “misclassifying fraudulent firms is 5 times worse than misclassifying solvent firms”)

Minimizing Cost Ratio

q_1 = cost of misclassifying an actual "1",

q_0 = cost of misclassifying an actual "0"

Minimizing the **cost ratio** q_1/q_0 is identical to minimizing the average cost per record

Software* may provide option for user to specify cost ratio

*Currently unavailable in XLMiner

Note: Opportunity costs

- As we see, best to convert everything to costs, as opposed to a mix of costs and benefits
- E.g., instead of “benefit from sale” refer to “opportunity cost of lost sale”
- Leads to same decisions, but referring only to costs allows greater applicability

Cost Matrix
(inc. opportunity costs)

	Predict as 1	Predict as 0
Actual 1	$8	$20
Actual 0	$20	$0

Recall original confusion matrix (profit from a "1" = $10, cost of sending offer = $1):

	Predict as 1	Predict as 0
Actual 1	8	2
Actual 0	20	970

Multiple Classes

For *m* classes, confusion matrix has *m* rows and *m* columns

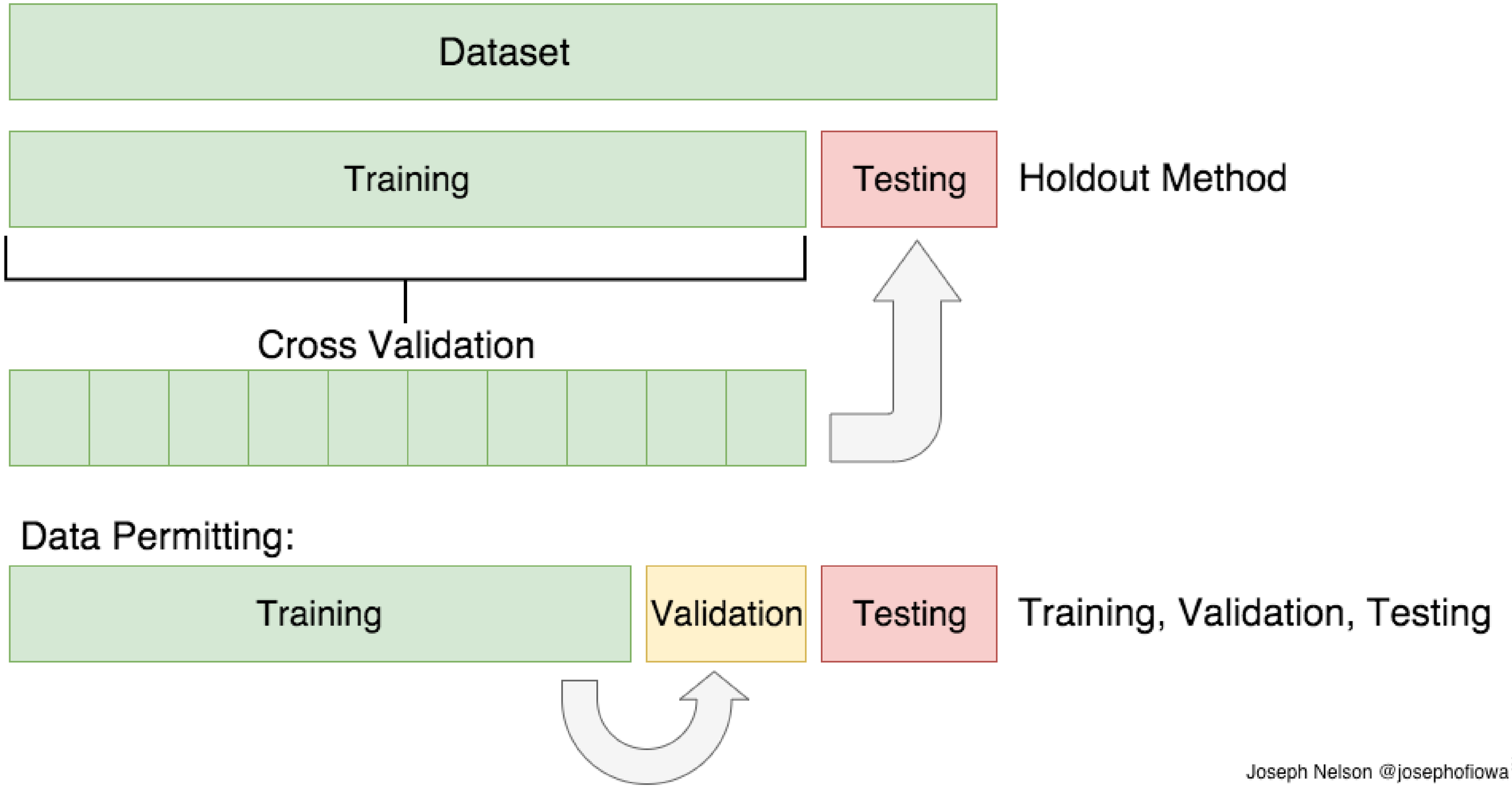
- Theoretically, there are *m*(*m*-1) misclassification costs, since any case could be misclassified in *m*-1 ways
- Practically too many to work with
- In decision-making context, though, such complexity rarely arises – one class is usually of primary interest

Confusion Matrix for Multi-class problems

https://www.analyticsvidhya.com/blog/2021/06/confusion-matrix-for-multi-class-classification/#:~:text=The%20confusion%20matrix%20is%20a,and%20False%20Negative(FN).

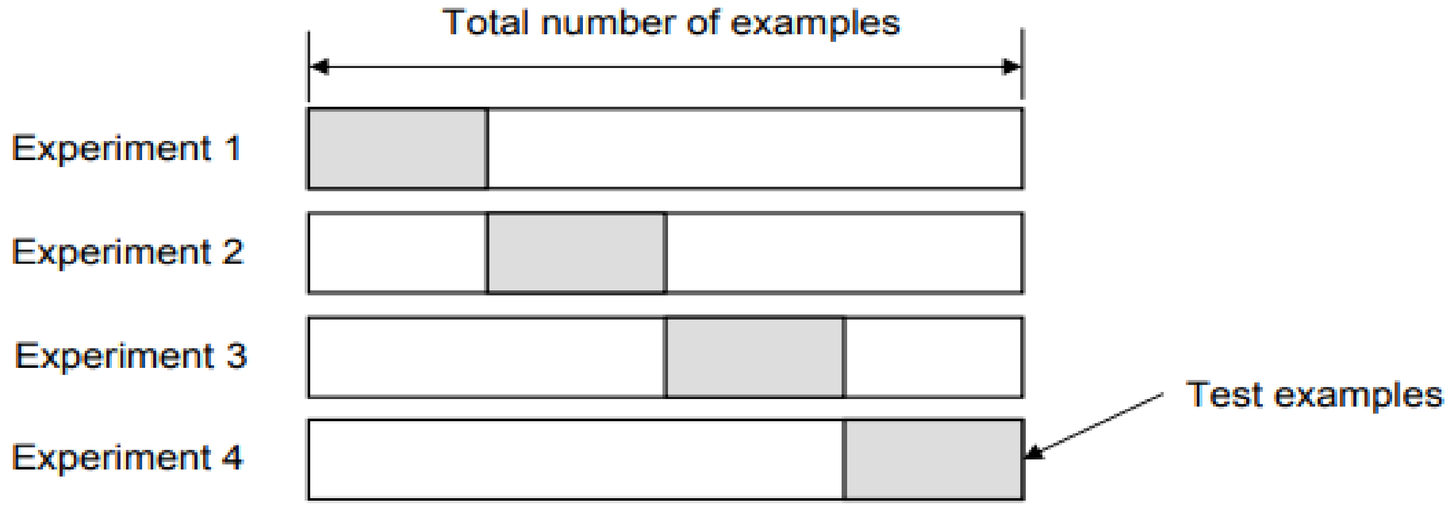
https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826

Dividing the dataset into training and testing sets



k-Fold Cross Validation

- **Create a K-fold partition of the the dataset**
  - For each of K experiments, use K-1 folds for training and a different fold for testing
    - This procedure is illustrated in the following figure for K=4



- **K-Fold Cross validation is similar to Random Subsampling**
  - The advantage of K-Fold Cross validation is that all the examples in the dataset are eventually used for both training and testing
- **As before, the true error is estimated as the average error rate on test examples**

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

Source: https://www.csd.uwo.ca/~oveksler/Courses/Winter2006/CS434_654b/pr_l13.pdf

Summary

- Evaluation metrics are important for comparing across DM models, for choosing the right configuration of a specific DM model, and for comparing to the baseline
- Major metrics: confusion matrix, error rate, predictive error
- Other metrics when
  - one class is more important
  - asymmetric costs
- When important class is rare, use oversampling
- In all cases, metrics computed from validation data

The content of the slides are prepared from different textbooks.

References:

- Data Mining and Predictive Analytics, By Daniel T. Larose. Copyright 2015 John Wiley & Sons, Inc.
- Predictive Analytics for Dummies, By Anasse Bari, Mohamed Chaouchi, & Tommy Jung, Copyright 2016, John Wiley & Sons, Inc.
- Introduction to Data Mining with Case Studies, By G.K. Gupta. Copyright 2014 by PHI Learning Private Limited.


Thank you..

Machine Learning with Python

Classification and Regression Trees

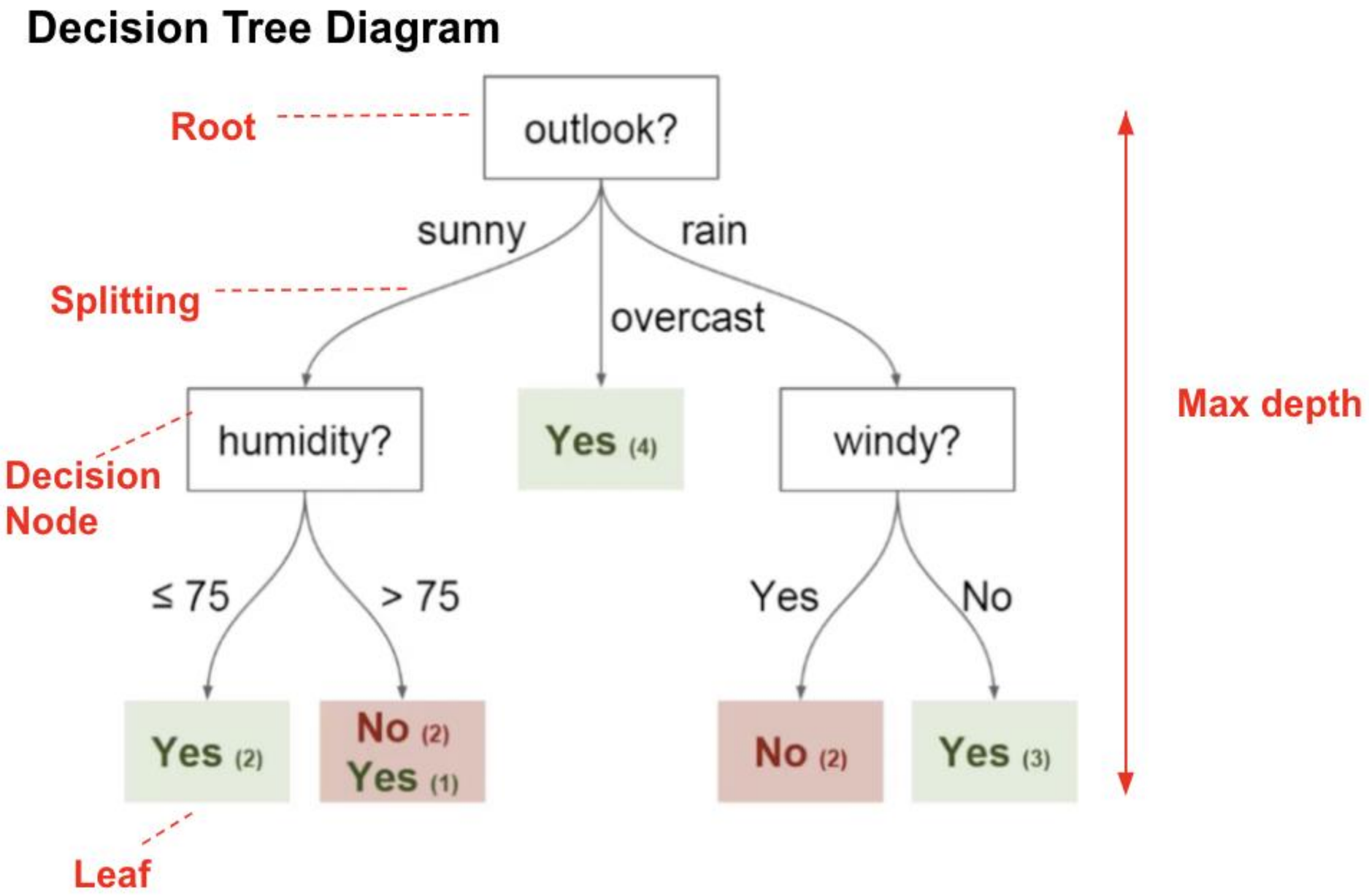
Arghya Ray

Decision Tree

- A decision tree is a popular classification method that results in a flow-chart like tree structure where each node denotes a test on an attribute value and each branch represents an outcome of the test. The tree leaves represent the classes.
- Decision tree is a model that is both predictive and descriptive.

- **Advantages:**
  - Decision tree approach is widely used since it is efficient and can deal with both continuous and categorical variables.
  - The decision tree approach is able to deal with missing values in the training data and can tolerate some errors in data.
  - The decision tree approach is perhaps the best if each attribute takes only a small number of possible values.
- **Disadvantages**:
  - Decision trees are less appropriate for tasks where the task is to predict values of a continuous variable like share price or interest rate.
  - Decision trees can lead to a large number of errors if the number of training examples per class is small.
  - The complexity of a decision tree increases as the number of attributes increases.

- ***Measuring the quality of a decision tree*** is an interesting problem altogether. ***Classification accuracy*** determined using test data is obviously a good measure but other measures like, ***average cost*** and ***worst case cost*** of classifying an object may be used.

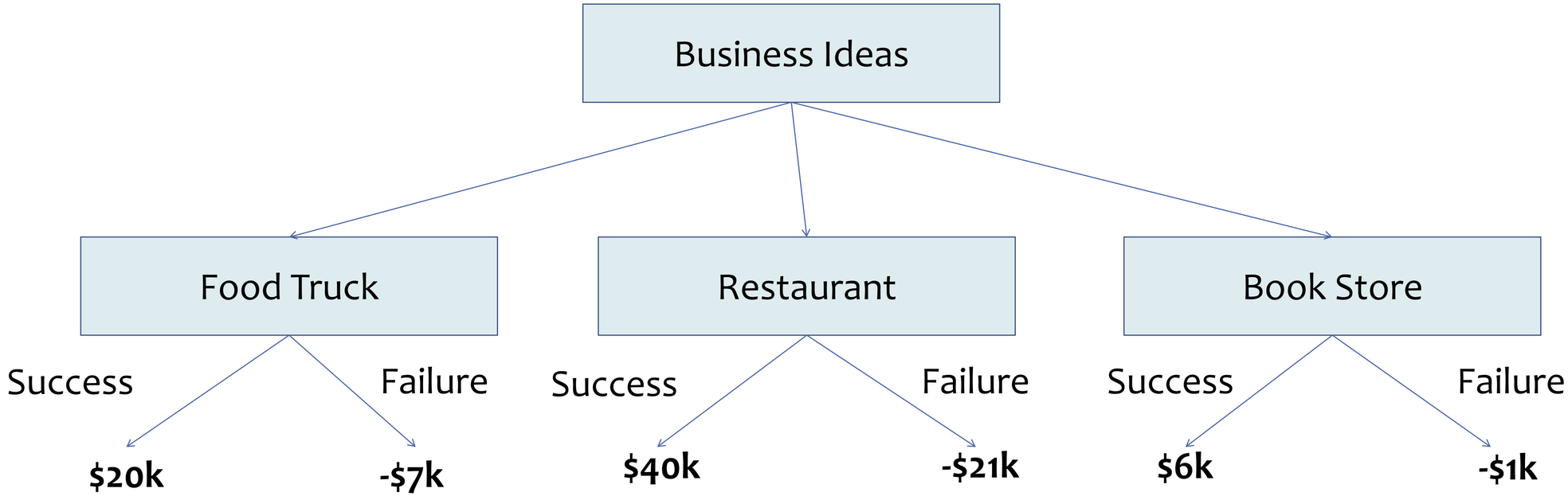


Picture taken from Velocity Business Solutions. Link: https://www.vebuso.com/2020/01/decision-tree-intuition-from-concept-to-application/

1. A decision tree is an approach to analysis that can help you make decisions.

Suppose for example you need to decide whether to invest a certain amount of money in one of the three business projects: a food-truck business, a restaurant, or a bookstore based on the data given below.

	Business Success Percentage		Business Value Changes	
Business	**Success Rate**	**Failure Rate**	**Gain (USD)**	**Loss (USD)**
Food Truck	60%	40%	20000	-7000
Restaurant	52%	48%	40000	-21000
Bookstore	50%	50%	6000	-1000



- In these cases, ***the expected value*** calculated based on all possible outcomes helps in figuring out the business decision making.

- Expected Value for the food truck business = (60% of USD 20000)+ (40% of USD (-7000)) = USD 9200.

- Expected Value of restaurant business = (52% of USD 40000) + (48% of USD (-21000)) = USD 10720.

- Expected Value of bookstore business = (50% of USD 6000) + (50% of USD (-1000)) = USD 2500

- Here the expected value reflects the average gain from investing in the business. Based on the above hypothetical figures, the results reflect that if you attempt to invest in a businesses say Food Truck business several times (under the same circumstances each time), your average profit will be USD 9200 per business.

2. Decision trees can also be used to visualize classification rules.

Classification and Regression Trees

Goal: Classify or predict an outcome based on a set of predictors.

The output is a set of **rules**

Example:

- Goal: classify a record as "will accept credit card offer" or "will not accept"
- Rule might be "IF (Income > 92.5) AND (Education < 1.5) AND (Family <= 2.5) THEN Class = 0 (non-acceptor)
- **Recursive partitioning:** Repeatedly split the records into two parts so as to achieve maximum homogeneity within the new parts

Recursive partitioning steps:

- Pick one of the predictor variables, xi
- Pick a value of xi, say si, that divides the training data into two (not necessarily equal) portions
- Measure how "pure" or homogeneous each of the resulting portions are
- "Pure" = containing records of mostly one class
- Algorithm tries with different variables (x) and different values of x_i, , i.e., s_i to maximize purity in a split
- After you get a "maximum purity" split, repeat the process for a second split, and so on

Forming a tree from the given example

RID	Age	Income	Student	Credit rating	Class (buys computer)
1	<=30	High	No	Fair	No
2	<=30	High	No	Excellent	No
3	31-40	High	No	Fair	Yes
4	>40	Medium	No	Fair	Yes
5	>40	Low	Yes	Fair	Yes
6	>40	Low	Yes	Excellent	No
7	31-40	Low	Yes	Excellent	Yes
8	<=30	Medium	No	Fair	No
9	<=30	Low	Yes	Fair	Yes
10	>40	Medium	Yes	Excellent	Yes
11	<=30	Medium	Yes	Excellent	Yes
12	30-40	Medium	No	Excellent	Yes
13	30-40	High	Yes	Fair	Yes
14	>40	Medium	No	Excellent	No

<=30

income	student	Credit rating	Class
High	No	fair	No
High	No	excellent	No
Medium	No	Fair	No
Low	Yes	Fair	Yes
Medium	Yes	Excellent	yes

31-40

income	student	Credit rating	Class
High	No	Fair	Yes
Low	Yes	Excellent	Yes
Medium	No	Excellent	Yes
High	Yes	Fair	Yes

>40

income	student	Credit rating	Class
Medium	No	fair	Yes
Low	Yes	Fair	Yes
Low	Yes	Excellent	No
Medium	Yes	Fair	Yes
Medium	No	Excellent	No

Measuring Impurity

- Gini Index (measure of impurity)
  - Gini Index for rectangle *A* containing *m* cases

$$I(A) = 1 - \sum_{k=1}^{m} p_k^2$$

p = proportion of cases in rectangle *A* that belong to class *k*

  - I(A) = 0 when all cases belong to same class (most pure)

- Entropy (measure of impurity)

$$entropy\,(A) = -\sum_{k=1}^{m} p_k \log_2(p_k)$$

p = proportion of cases (out of *m*) in rectangle *A* that belong to class *k*

Entropy ranges between 0 (most pure) and $\log_2(m)$ (equal representation of classes)

Using the principle of ‘Information entropy’ build a ‘decision tree’ using the training data given below. Divide the ‘credit rating’ attribute into ranges as follows: (0, 1.6], (1.6,1.7], (1.7,1.8], (1.8,1.9], (1.9,2.0], (2.0,5.0]

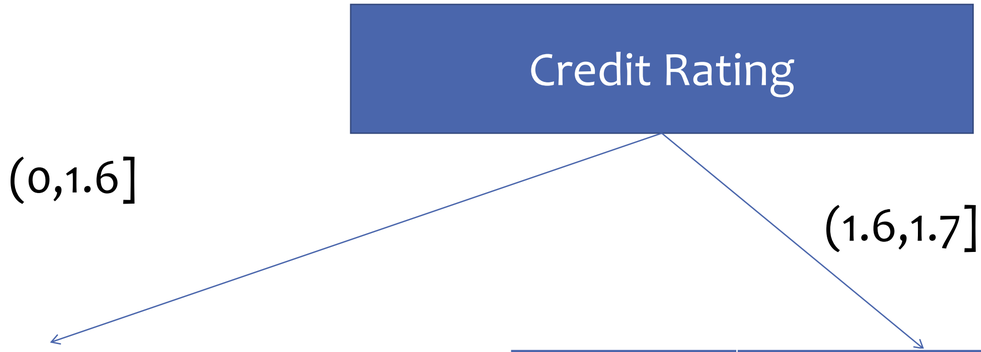
Sr. No.	Profession	Credit rating	Class
1	Business	1.6	Buys only laptop
2	Service	2.0	Buys laptop with CD Writer
3	Business	1.9	Buys laptop with printer
4	Business	1.88	Buys laptop with printer
5	Business	1.70	Buys only laptop
6	Service	1.85	Buys laptop with printer
7	Business	1.60	Buys only laptop
8	Service	1.70	Buys only laptop
9	Service	2.20	Buys laptop with CD writer
10	Service	2.10	Buys laptop with CD writer
11	Business	1.80	Buys laptop with printer
12	Service	1.95	Buys laptop with printer
13	Business	1.90	Buys laptop with printer
14	Business	1.80	Buys laptop with printer
15	Business	1.75	Buys laptop with printer

Business

Service

Sr. No.	Credit rating	Class
1	1.6	Buys only laptop
2	1.9	Buys laptop with printer
3	1.88	Buys laptop with printer
4	1.70	Buys only laptop
5	1.60	Buys only laptop
6	1.80	Buys laptop with printer
7	1.90	Buys laptop with printer
8	1.80	Buys laptop with printer
9	1.75	Buys laptop with printer

Sr. No.	Credit rating	Class
1	2.0	Buys laptop with CD Writer
2	1.85	Buys laptop with printer
3	1.70	Buys only laptop
4	2.20	Buys laptop with CD writer
5	2.10	Buys laptop with CD writer
6	1.95	Buys laptop with printer



Sr. No.	Profession	Class
1	Business	Buys only Laptop
2	Business	Buys only Laptop

Sr. No.	Profession	Class
1	Business	Buys only Laptop
2	Service	Buys only Laptop

Initially there are 3 classes: Buys only laptop, buys laptop with CD writer, buys laptop with printer

Initial Overall Entropy (E_o)= $-\sum_{i=1}^{3} p_i log_3 p_i = -[\frac{4}{15} log_3 \frac{4}{15} + \frac{3}{15} log_3 \frac{3}{15} + \frac{8}{15} log_3 \frac{8}{15}) = 0.918$

Based on Profession : 9 Business, 6 Service

Entropy (Profession) = $\frac{9}{15}$ $Entropy(business) + \frac{6}{15}$ $Entropy$ (service) = $\frac{9}{15}\left(-\frac{3}{9} log_3 \frac{3}{9} - \frac{6}{9} log_3 \frac{6}{9}\right) + \frac{6}{15}\left(-\frac{1}{6} log_3 \frac{1}{6} - \frac{2}{6} log_3 \frac{2}{6} - \frac{3}{6} log_3 \frac{3}{6}\right) = 0.71582$

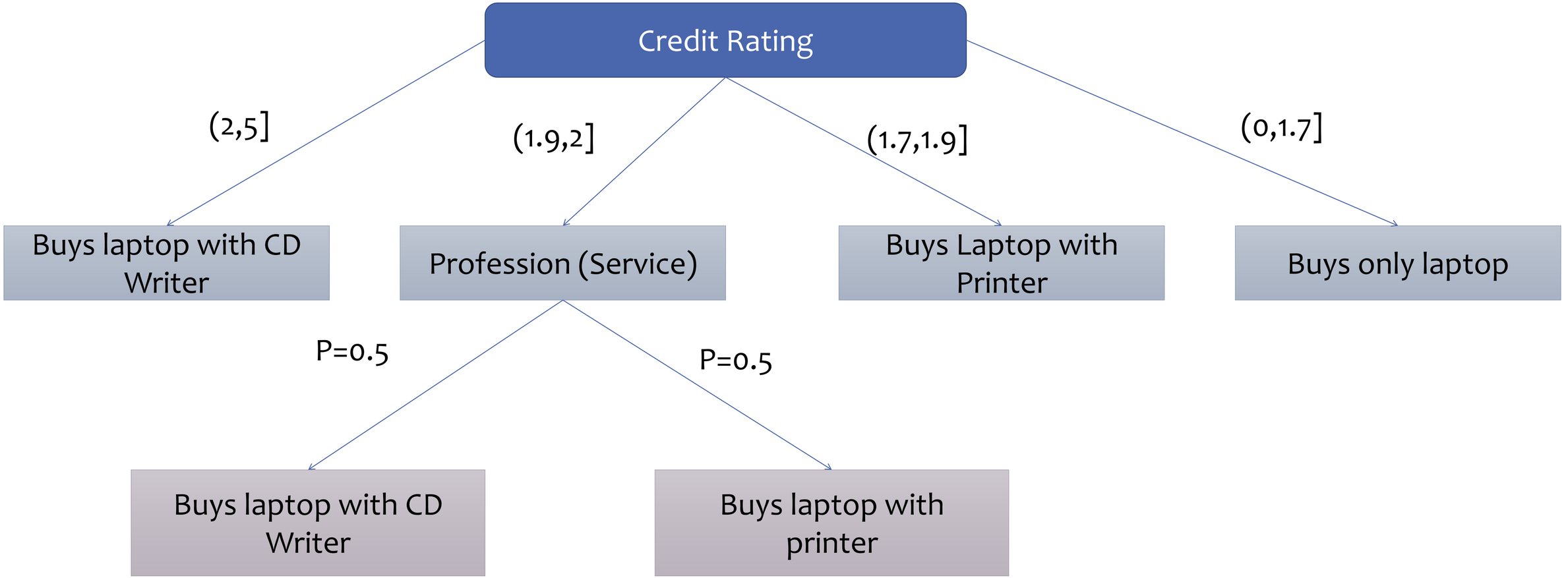
Information Gain (Profession) = E0-E(Profession) = 0.918-0.716= 0.202

Entropy (CR (2,5])=Entropy(CR (0, 1.6])= Entropy (CR (1.6,1.7]) = Entropy (CR (1.7,1.8]) = Entropy(CR (1.8,1.9]) = 0

Entropy (CR (1.9,2])= $-\frac{1}{2} log_3 \frac{1}{2} - \frac{1}{2} log_3 \frac{1}{2}$ = 0.630

Entropy (Credit Rating) = $\frac{2}{15}$ Entropy (CR (2,5])+$\frac{2}{15}$ Entropy (CR (1.9,2])+$\frac{3}{15}$ Entropy (CR (1.7,1.8])+$\frac{4}{15}$ Entropy (CR (1.8,1.9])+$\frac{2}{15}$ Entropy (CR (0,1.6]) +$\frac{2}{15}$ Entropy (CR (1.6,1.7]) = 0.0841

Information Gain (Credit Rating) = 0.918-0.084 = 0.834


Credit Rating
(2,5]
(1.9,2]
(1.7,1.9]
(0,1.7]
Buys laptop with CD Writer
Profession (Service)
Buys Laptop with Printer
Buys only laptop
P=0.5
P=0.5
Buys laptop with CD Writer
Buys laptop with printer

The content of the slides are prepared from different textbooks.

References:

- Data Mining and Predictive Analytics, By Daniel T. Larose. Copyright 2015 John Wiley & Sons, Inc.
- Predictive Analytics for Dummies, By Anasse Bari, Mohamed Chaouchi, & Tommy Jung, Copyright 2016, John Wiley & Sons, Inc.
- Introduction to Data Mining with Case Studies, By G.K. Gupta. Copyright 2014 by PHI Learning Private Limited.


Thank you..

Machine Learning with Python

Ensemble Method (Bagging and Boosting)

Arghya Ray

Ensemble Methods: Bagging and Boosting

Rationale

- The ensemble classifier is likely to have a lower error rate (boosting)
- The variance of the ensemble classifier will be lower than had we used certain unstable classification models (such as decision trees and neural nets) that have high variability (bagging and boosting)
- Suppose we have an ensemble of binary classifiers each with error rate 0.20. If the individual classifiers agree the error rate will be the same as for the individual classifiers, the ensemble classifier will make an error only when the majority of individual classifiers make an error

Rationale (cont.)

Let ε represent the individual classifier error rate. The probability that *k* of the 5 individual classifiers will make the same wrong prediction is:

$$\binom{5}{k}\epsilon^k(1-\epsilon)^{5-k} = \binom{5}{k}0.2^k(1-0.2)^{5-k}$$

And the probability that all 5 of the classifiers will make an error is:

$$\binom{5}{5}0.2^5(0.8)^0 = 0.00032$$

And the error rate for the ensemble is:

$$Error\ rate_{\ Ensemble\ classifier} = \sum_{i=3}^{5}\binom{5}{i}\epsilon^i(1-\epsilon)^{5-i} = \sum_{i=3}^{5}\binom{5}{i}0.2^i(0.8)^{5-i}$$
$$= 0.0512 + 0.0064 + 0.0003 = 0.05792$$

Bias, Variance, and Noise

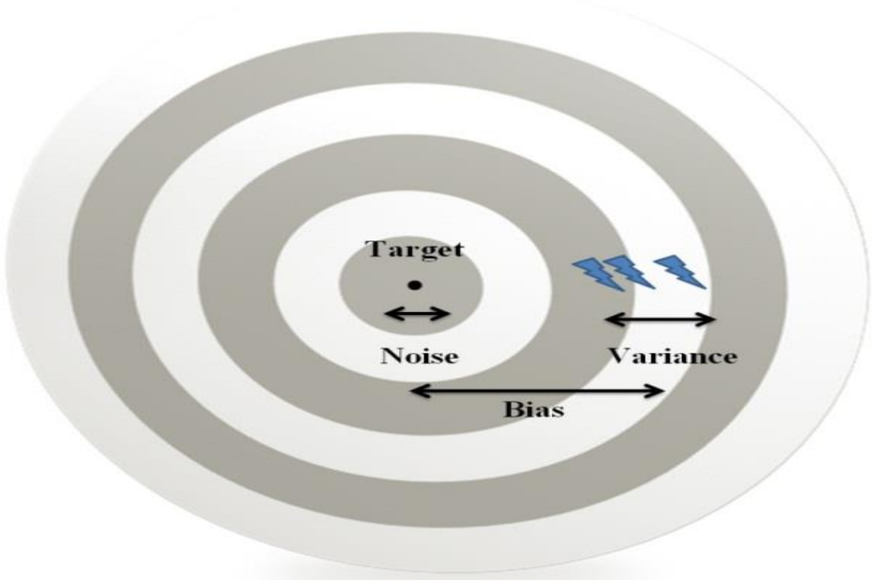
- We would like our models to have a low prediction error, which can be decomposed as:

$$(y - \hat{y}) = Bias + Variance + Noise$$

- Where *Bias* refers to the average distance between the predictions ($\hat{y}$, represented by the lightning darts in Figure 25.1) and the target (y, the bull's eye),

- And *Variance* measures the variability in the predictions $\hat{y}$ themselves, and

- And *Noise* represents the lower bound on the prediction error that the predictor can possibly achieve.



Bias, Variance, and Noise (cont.)

- To reduce prediction error we need to reduce the bias, variance or noise. Unfortunately noise is an intrinsic characteristic of the prediction problem and can not be reduced.
- Bagging can reduce the variance of the classifier models
- Boosting can reduce both bias and variance and thus offers a way to short-circuit the *bias-variance tradeoff*, where efforts to reduce bias necessarily increase the variance and vice-versa.

When to Apply Bagging

"Some classification and regression methods are unstable in the sense that small perturbations in their training sets or in construction may result in large changes in the constructed predictor." (Breiman 1998)

Classification Algorithm	Stable or Unstable
Classification and Regression Trees	Unstable
C4.5	Unstable
Neural Networks	Unstable
k-Nearest Neighbor	Stable
Discriminant Analysis	Stable
Naïve Bayes	Stable

When to Apply Bagging (cont.)

- Bagging works best with unstable models where there is room for improvement in reducing variability as it is a method for reducing variance.

- Applying bagging to stable models can degrade their performance

- Bagging works with bootstrap samples of the original data, each of which contains only about 63% of the data

Bagging

Bagging coined by Leo Breiman to refer to the following *Boostrap Aggregating* Algorithm:

1. Samples (with replacement) are repeatedly taken from the training data set, so that each record has an equal probability of being selected, and each sample is the same size as the original training data set. These are the bootstrap samples.

2. A classification or estimation model is trained on each bootstrap sample drawn in Step 1, and a prediction is recorded for each sample.

3. The bagging ensemble prediction is then defined to be the class with the most votes in Step 2 (for classification models) or the average of the predictions made in Step 2 (for estimation models).

Bagging Example

- Consider a small data set in which x is the variable value and y is the classification.

x	0.2	0.4	0.6	0.8	1
y	1	0	0	0	1

- We have a one-level decision tree classifier that chooses a value of *k* to minimize leaf node entropy

- If bagging is not used the best the classifier can do is a 20% error rate with a split at $x \leq 0.3$ or $x \leq 0.9$

Bagging Example (cont.)

Step 1: Bootstrap samples are taken, and Step 2: one-level decision tree classifiers (base classifiers) are trained on each sample.

Bootstrap Sample							Base Classifier
1	x	0.2	0.2	0.4	0.6	1	$x \leq 0.3 \Rightarrow y = 1$
	y	1	1	0	0	1	$otherwise\ y = 0$
2	x	0.2	0.4	0.4	0.6	0.8	$x \leq 0.3 \Rightarrow y = 1$
	y	1	0	0	0	0	$otherwise\ y = 0$
3	x	0.4	0.4	0.6	0.8	1	$x \leq 0.9 \Rightarrow y = 0$
	y	0	1	0	0	1	$otherwise\ y = 1$
4	x	0.2	0.6	0.8	1	1	$x \leq 0.9 \Rightarrow y = 0$
	y	1	0	0	1	1	$otherwise\ y = 1$
5	x	0.2	0.2	1	1	1	$x \leq 0.1 \Rightarrow y = 0$
	y	1	1	1	1	1	$otherwise\ y = 1$

Bagging Example (cont.)

Step 3. For each record tally votes and select the majority class. Since we have 0/1 classifier the majority is the proportion of 1's. If the proportion is less than 0.5 then the bagging prediction is 0, otherwise 1.

Bootstrap Sample	$x = 0.2$	$x = 0.4$	$x = 0.6$	$x = 0.8$	$x = 1$
1	1	0	0	0	0
2	1	0	0	0	0
3	0	0	0	0	1
4	0	0	0	0	1
5	1	1	1	1	1
Proportion	0.6	0.2	0.2	0.2	0.6
Bagging Prediction	1	0	0	0	1

Boosting

Boosting is an *adaptive* algorithm developed by Freund and Schapire in 1990's and reduces both error due to variance and error due to bias:

1. All observations have equal weight in the original training data set D_1. An initial "base" classifier h_1 is determined.
2. The observations that were incorrectly classified by the previous base classifier have their weights increased, while the observations that were correctly classified have their weights decreased. This gives us data distribution $D_m, m = 2, \dots, M$. A new base classifier $h_m, m = 2, \dots, M$ is determined, based on the new weights. This step is repeated until the desired number of iterations M is achieved.
3. The final boosted classifier is the weighted sum of the M base classifiers.

Boosting Step 1:

Training data D_1 consists of 10 dichotomous values as shown below. An initial base classifier h_1 is determined to separate the two leftmost values. Shaded area represents values classified as "+". Boxed values are those incorrectly classified.

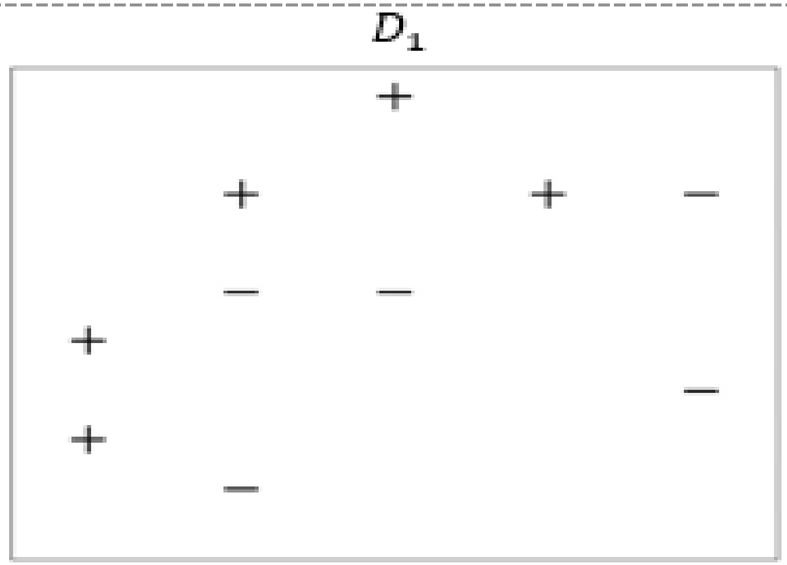


Figure 25.2 Original data

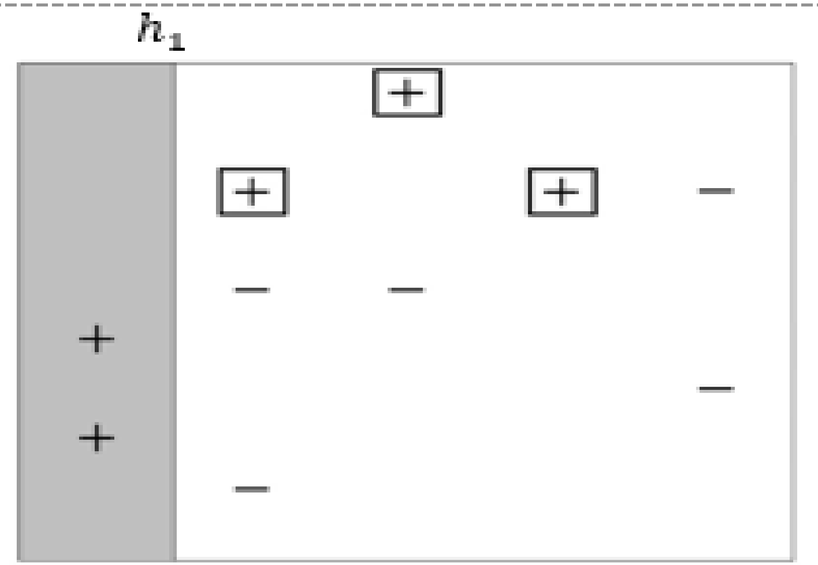


Figure 25.3 Initial base classifier

Boosting Step 2 (first pass):

3 values incorrectly classified by h_1 have weights (represented by relative size in diagrams) increased while other 7 have weights decreased. Based on the new weights a new classifier h_2 is determined

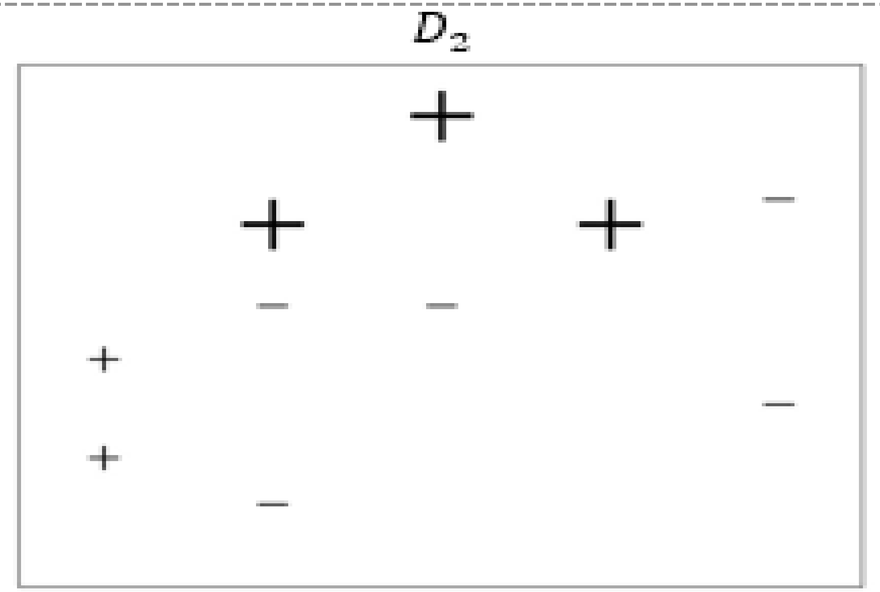


Figure 25.4 First reweighting of the data.

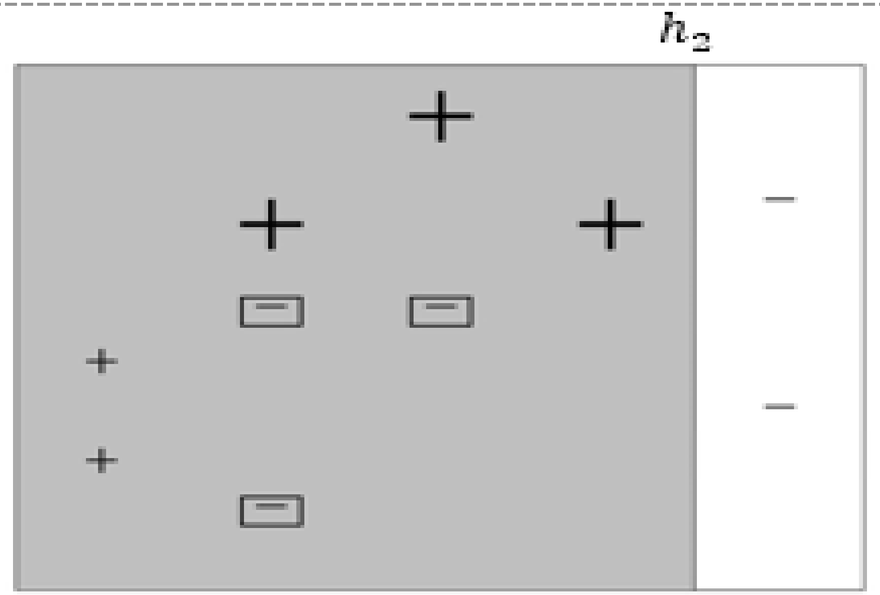


Figure 25.5 Second base classifier.

Boosting Step 2 (second pass):

3 values incorrectly classified by h_2 have weights increased while other 7 have weights decreased. Based on the new weights a new classifier h_3 is determined

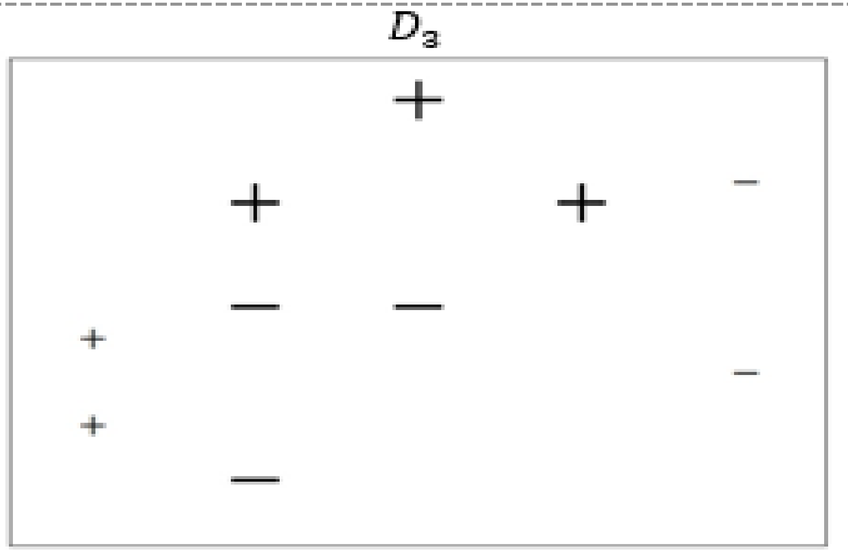


Figure 25.6 Second reweighting of the data.

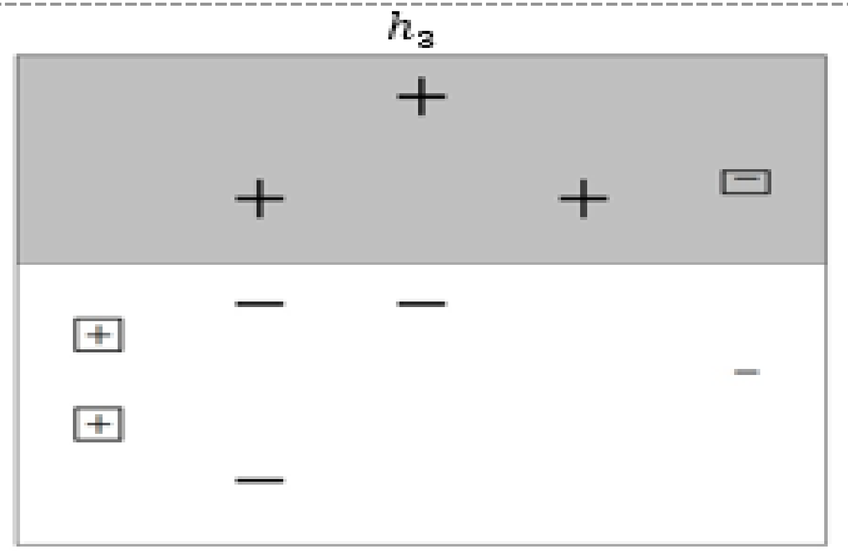
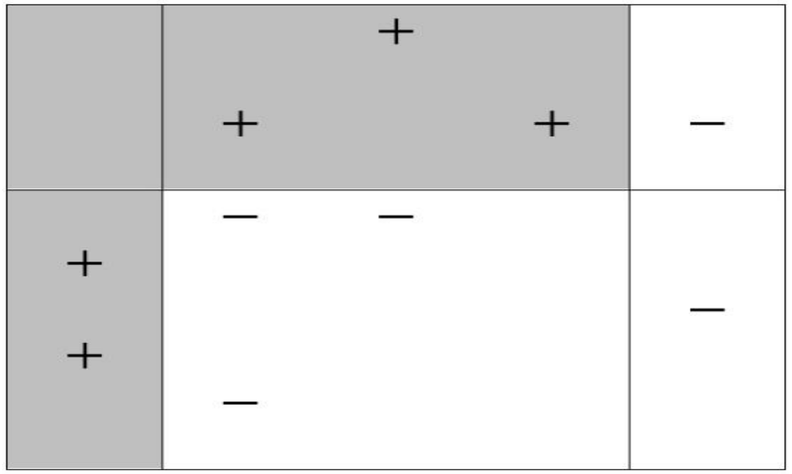


Figure 25.7 Third base classifier.

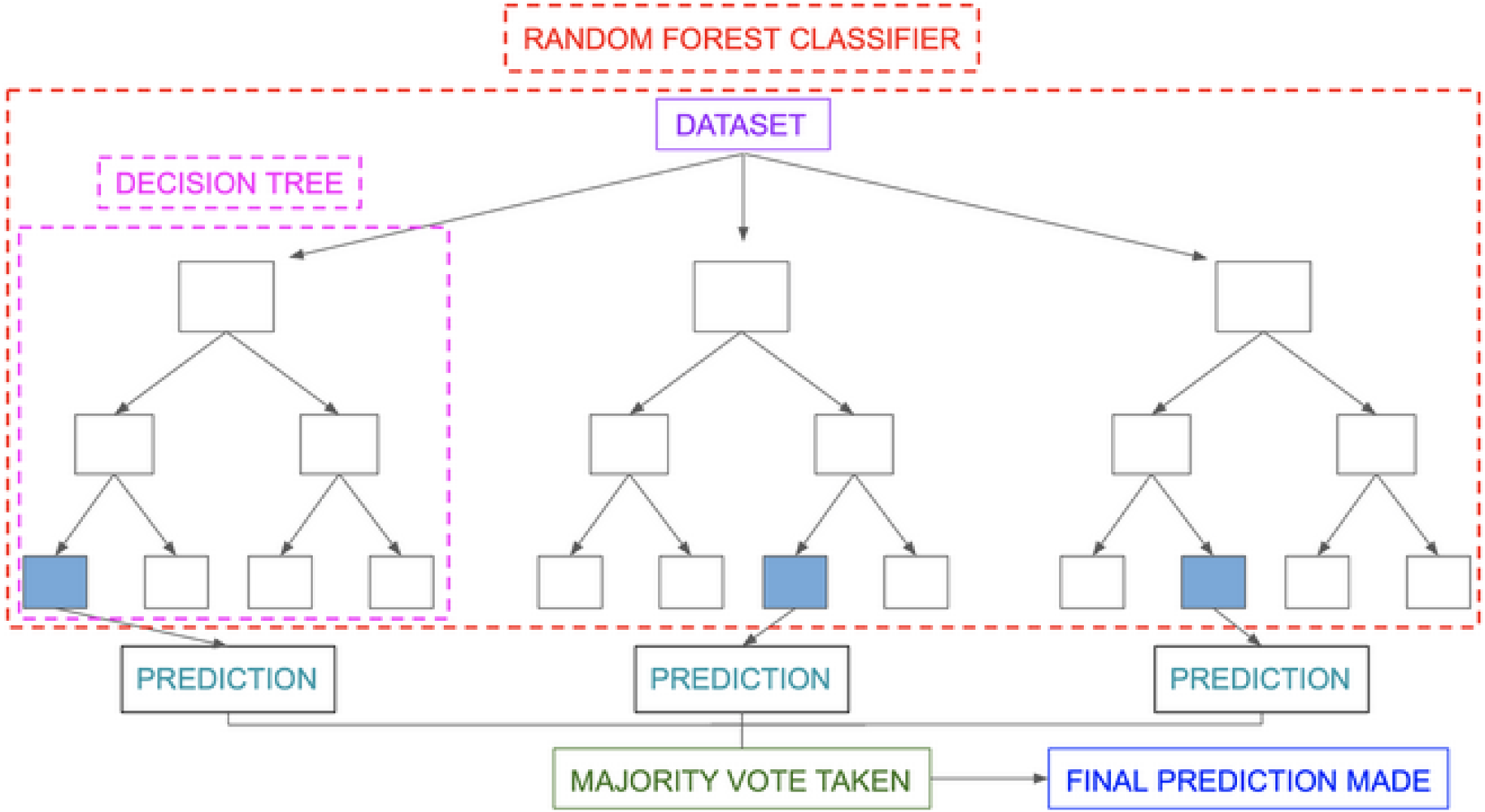
Boosting Step 3:

Final boosted classifier is the weighted sum of the $M = 3$ base classifiers: $\alpha_1 h_1 + \alpha_2 h_2 + \alpha_3 h_3$.



- Weights assigned each classifier proportional to accuracy of classifier
- Boosting performs best when base classifiers area unstable
- Boosting can increase the variance when the base classifier is stable

Random Forest



Good Article: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

https://towardsdatascience.com/random-forest-3a55c3aca46d

Classification in random forests:

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of observations and features that will be selected randomly during the splitting of nodes.

A rain forest system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this case, the output chosen by the majority of the decision trees becomes the final output of the rain forest system.

Regression in random forests

Regression is the other task performed by a random forest algorithm. A random forest regression follows the concept of simple regression. Values of dependent (features) and independent variables are passed in the random forest model.

In a random forest regression, each tree produces a specific prediction. The mean prediction of the individual trees is the output of the regression. This is contrary to random forest classification, whose output is determined by the mode of the decision trees' class.

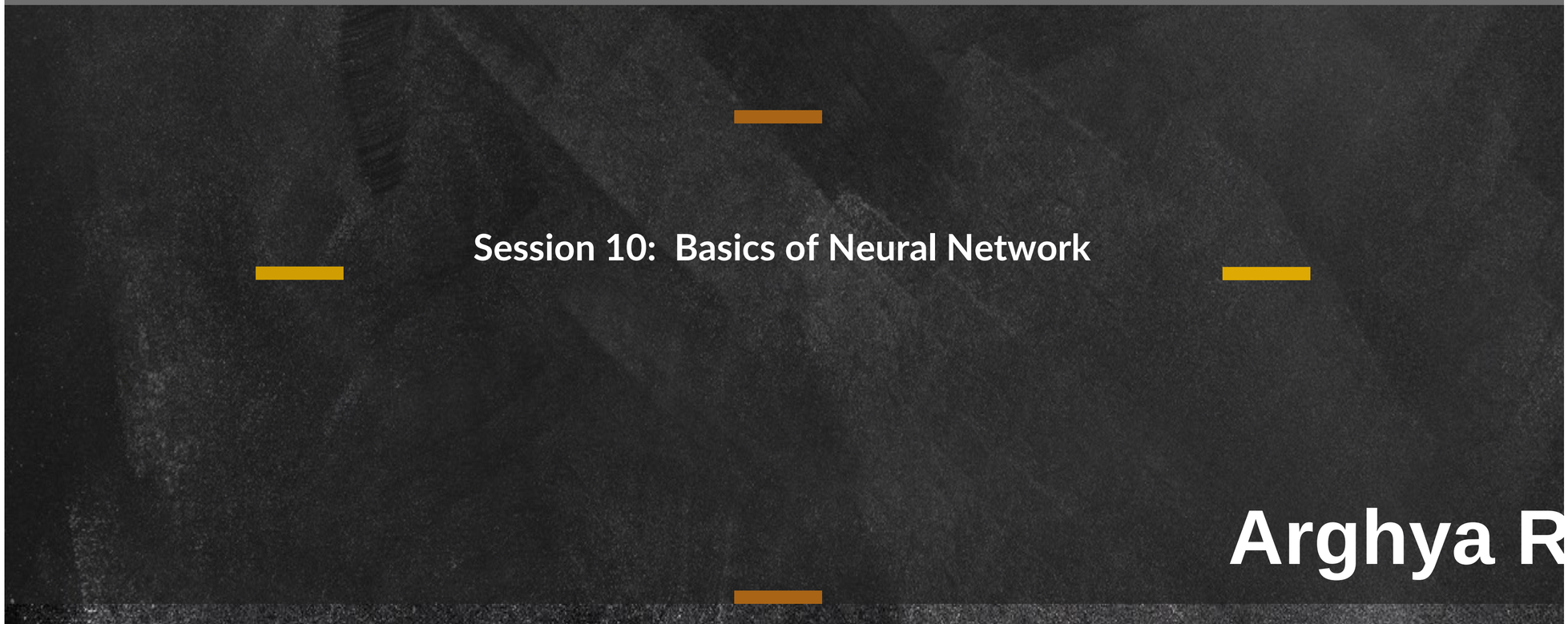
The content of the slides are prepared from different textbooks.

References:

- Data Mining and Predictive Analytics, By Daniel T. Larose. Copyright 2015 John Wiley & Sons, Inc.
- Predictive Analytics for Dummies, By Anasse Bari, Mohamed Chaouchi, & Tommy Jung, Copyright 2016, John Wiley & Sons, Inc.
- Introduction to Data Mining with Case Studies, By G.K. Gupta. Copyright 2014 by PHI Learning Private Limited.


Thank you..

Machine Learning with Python



Session 10: Basics of Neural Network

Arghya R

Neural Network for Classification

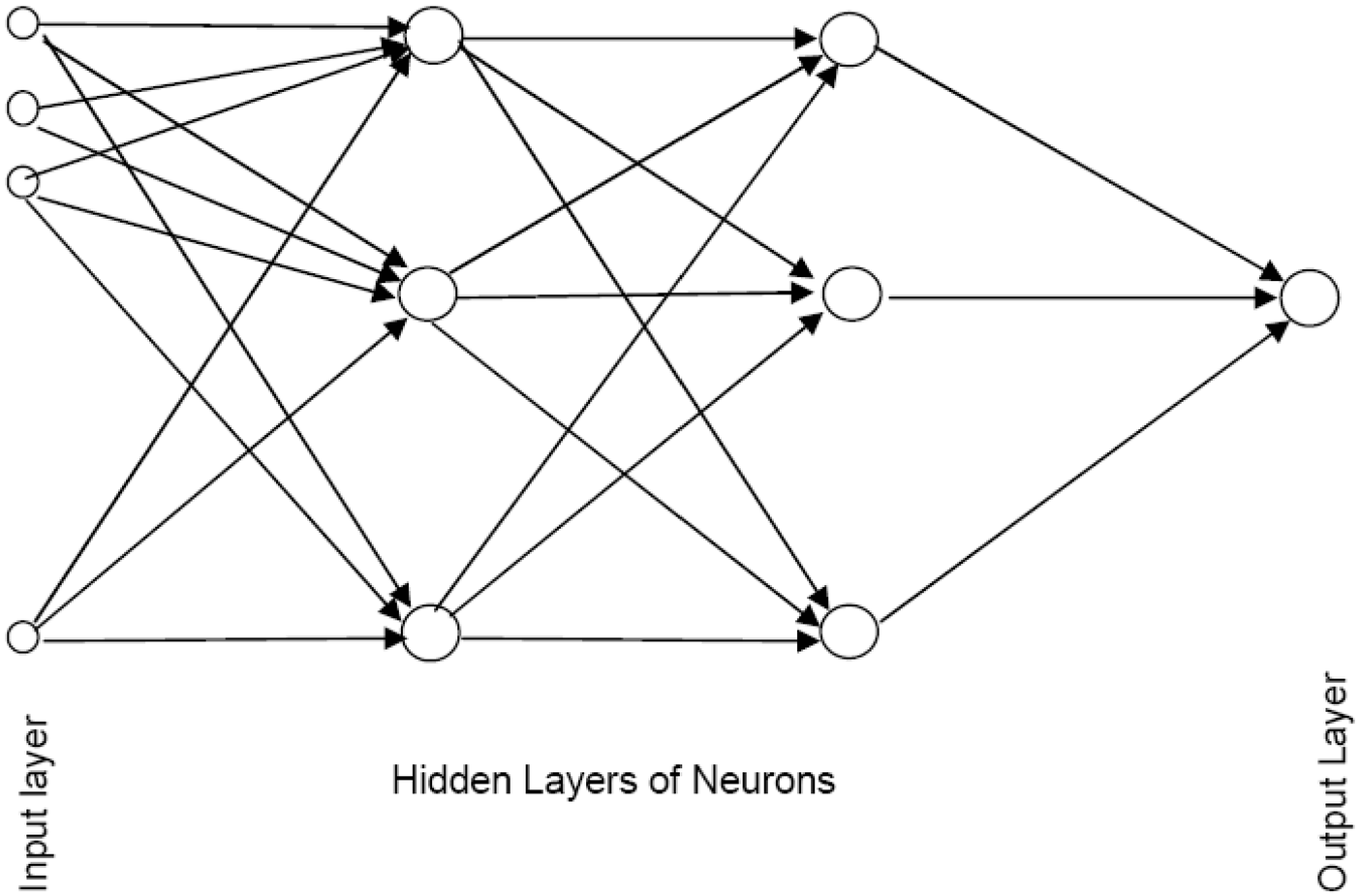
Basic Idea

- Combine input information in a complex & flexible neural net “model”
- Model “coefficients” are continually tweaked in an iterative process
- The network’s interim performance in classification and prediction informs successive tweaks

Network Structure

- Multiple layers
  - Input layer (raw observations)
  - Hidden layers
  - Output layer
- Nodes
- Weights (like coefficients, subject to iterative adjustment)
- Bias values (also like coefficients, but not subject to iterative adjustment)

Schematic Diagram



Example – Using fat & salt content to predict consumer acceptance of cheese

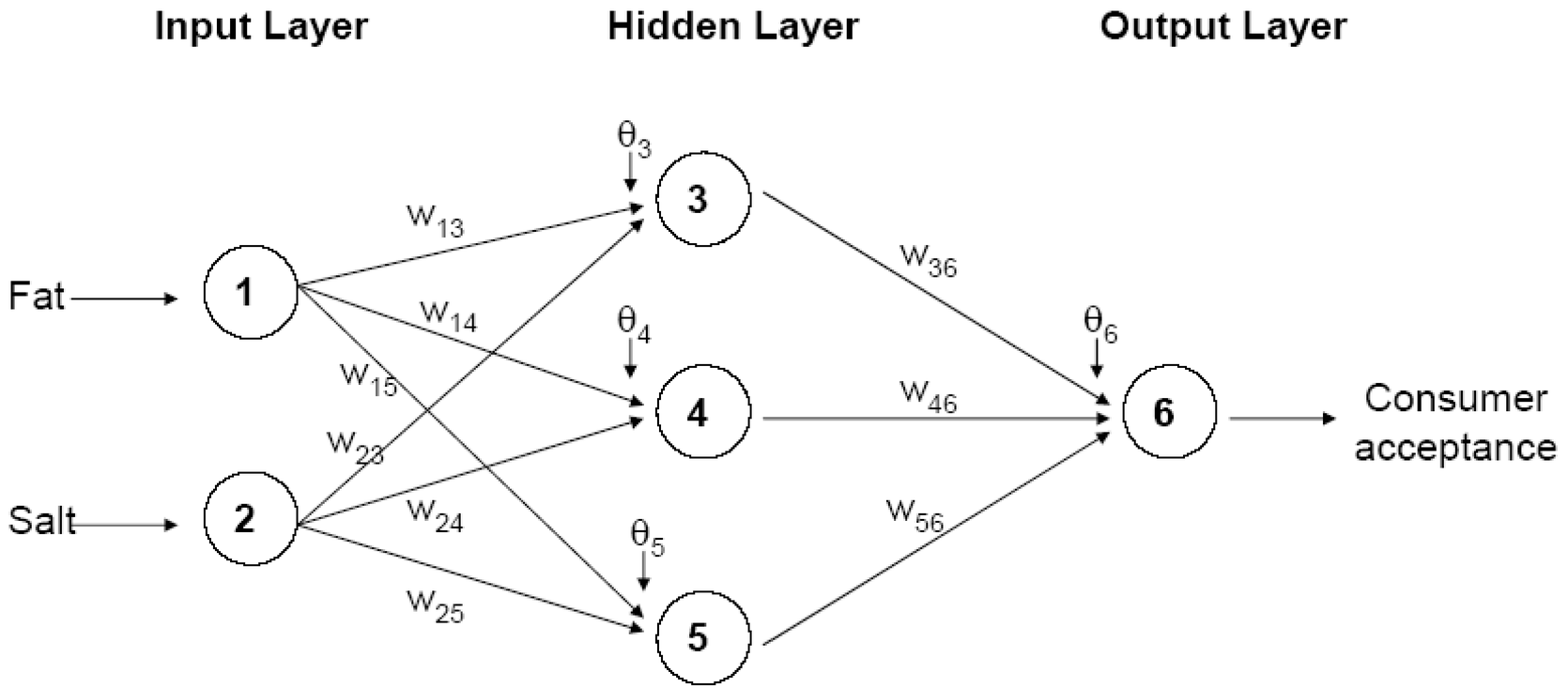


Figure 11.2: Neural network for the tiny example. Circles represent nodes, $w_{i,j}$ on arrows are weights, and θ_j are node bias values.

Example - Data

Obs.	*Fat Score*	*Salt Score*	*Acceptance*
1	0.2	0.9	1
2	0.1	0.1	0
3	0.2	0.4	0
4	0.2	0.5	0
5	0.4	0.5	1
6	0.3	0.8	1

Moving Through the Network

The Input Layer

For input layer, input = output

- E.g., for record #1:

  Fat input = 0.2

  Salt input = 0.9

Output of input layer = input into hidden layer

The Hidden Layer

In this example, hidden layer has 3 nodes

Each node receives as input the output of all input nodes

Output of each hidden node is a function of the weighted sum of inputs

$$output_j = g(\Theta_j + \sum_{i=1}^{p} w_{ij}\, x_i)$$

The Weights

The weights θ(theta) and *w* are typically initialized to random values in the range -0.05 to +0.05

Equivalent to a model with random prediction (in other words, no predictive value)

These initial weights are used in the first round of training

Output of Node 3, if g is a Logistic Function

$$output_j = g(\Theta_j + \sum_{i=1}^{p} w_{ij}\, x_i)$$

$$output_3 = \frac{1}{1 + e^{-[-0.3+(0.05)(0.2)+(0.01)(0.9)]}} = 0.43$$

Initial Pass of the Network

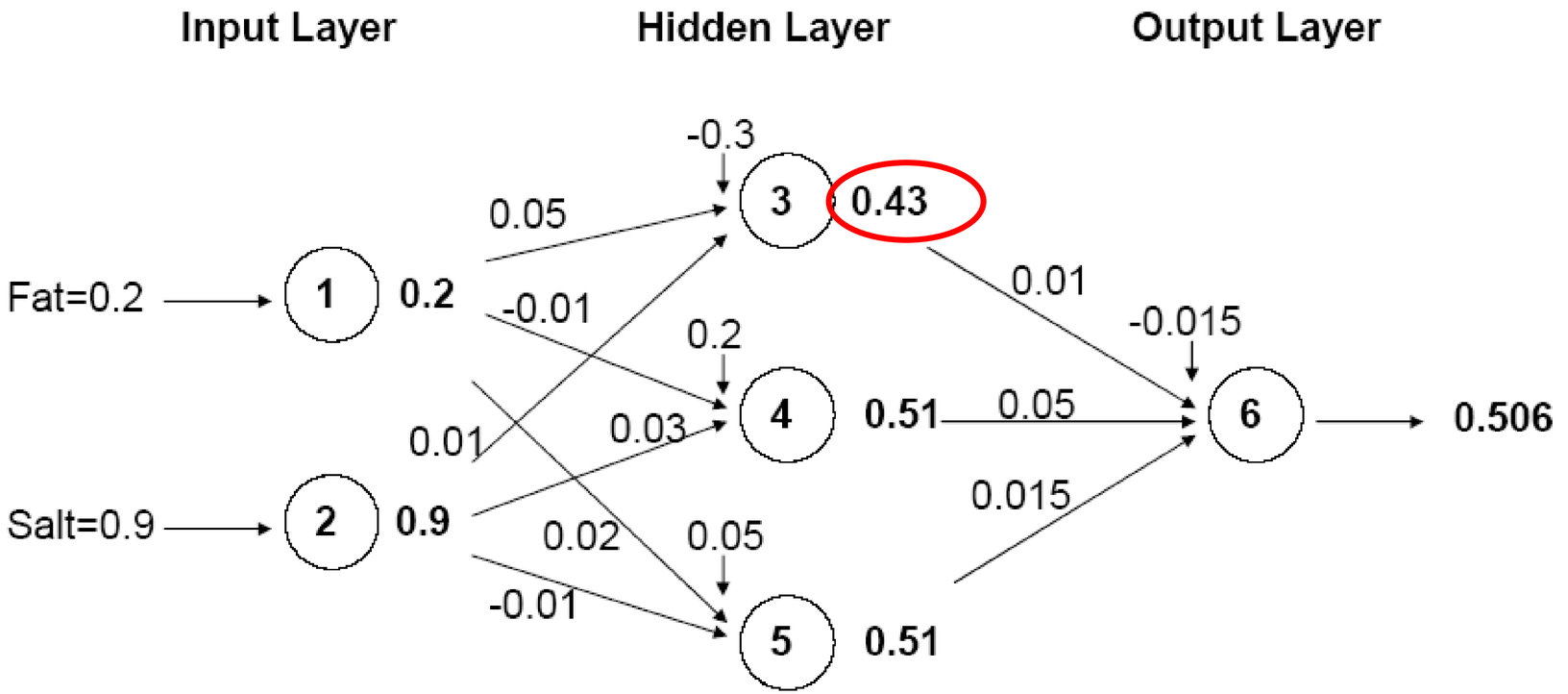


Figure 11.3: Computing node outputs (in boldface type) using the first observation in the tiny example and a logistic function.

Output Layer

The output of the last hidden layer becomes input for the output layer

Uses same function as above, i.e. a function g of the weighted average

$$output_6 = \frac{1}{1 + e^{-[-0.015+(0.01)(0.43)+(0.05)(0.507)+(0.015)(0.511)]}} = 0.506$$

The output node

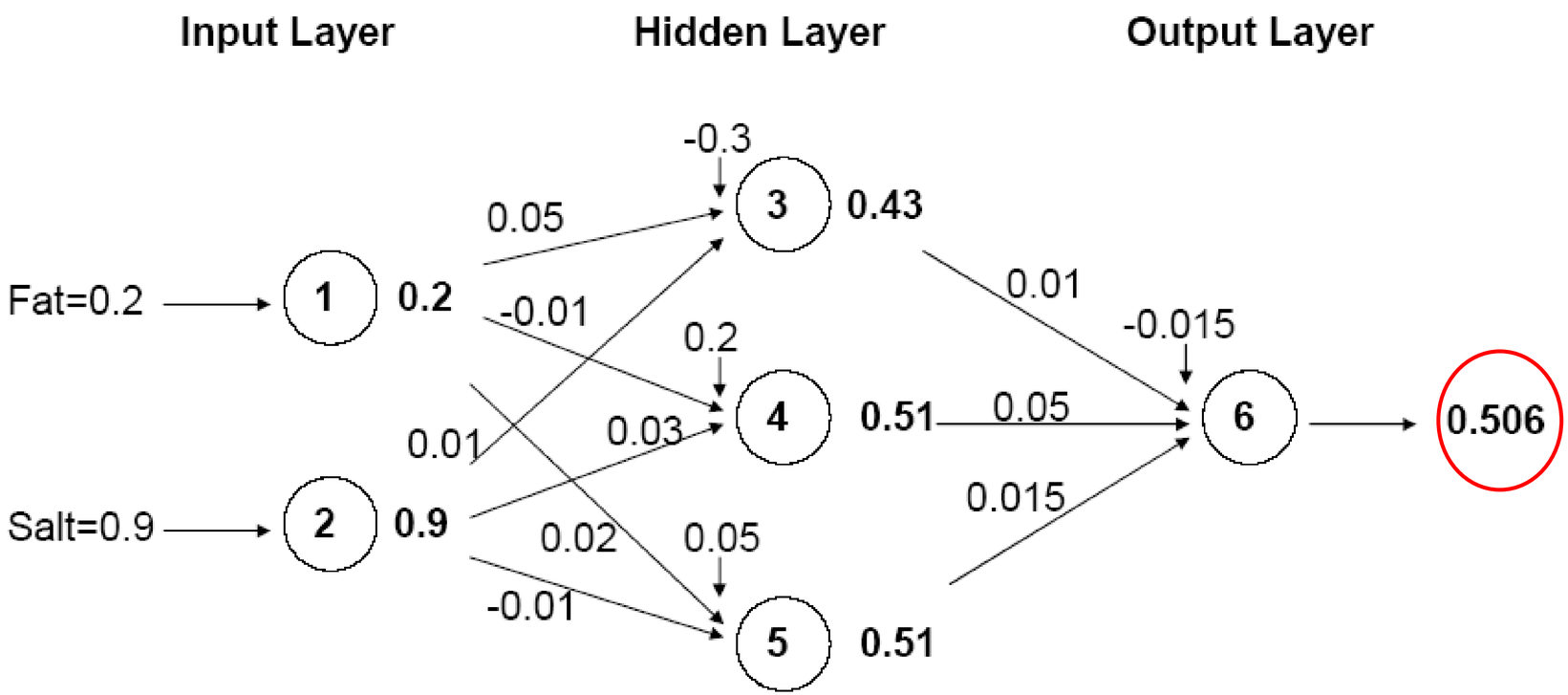


Figure 11.3: Computing node outputs (in boldface type) using the first observation in the tiny example and a logistic function.

Mapping the output to a classification

Output = 0.506

If cutoff for a “1” is 0.5, then we classify as “1”

Relation to Linear Regression

A net with a single output node and no hidden layers, where g is the identity function, takes the same form as a linear regression model

$$\hat{y} = \Theta + \sum_{i=1}^{p} w_i x_i$$

Training the Model

Preprocessing Steps

- Scale variables to 0-1
- Categorical variables
- If equidistant categories, map to equidistant interval points in 0-1 range
- Otherwise, create dummy variables
- Transform (e.g., log) skewed variables

Initial Pass Through Network

Goal: Find weights that yield best predictions

- The process we described above is repeated for all records
- At each record, compare prediction to actual
- Difference is the error for the output node
- Error is propagated back and distributed to all the hidden nodes and used to update their weights

Back Propagation (“back-prop”)

- Output from output node k: $\hat{y}_k$
- Error associated with that node:

$$err_k = \hat{y}_k(1 - \hat{y}_k)(y_k - \hat{y}_k)$$

Note: this is like ordinary error, multiplied by a correction factor

Error is Used to Update Weights

$$\theta_j^{new} = \theta_j^{old} + l(err_j)$$

$$\mathrm{w}_j^{new} = w_j^{old} + l(err_j)$$

l = constant between 0 and 1, reflects the "learning rate" or "weight decay parameter"

Case Updating

- Weights are updated after each record is run through the network
- Completion of all records through the network is one *epoch* (also called *sweep* or *iteration*)
- After one epoch is completed, return to first record and repeat the process

Batch Updating

- All records in the training set are fed to the network before updating takes place
- In this case, the error used for updating is the sum of all errors from all records

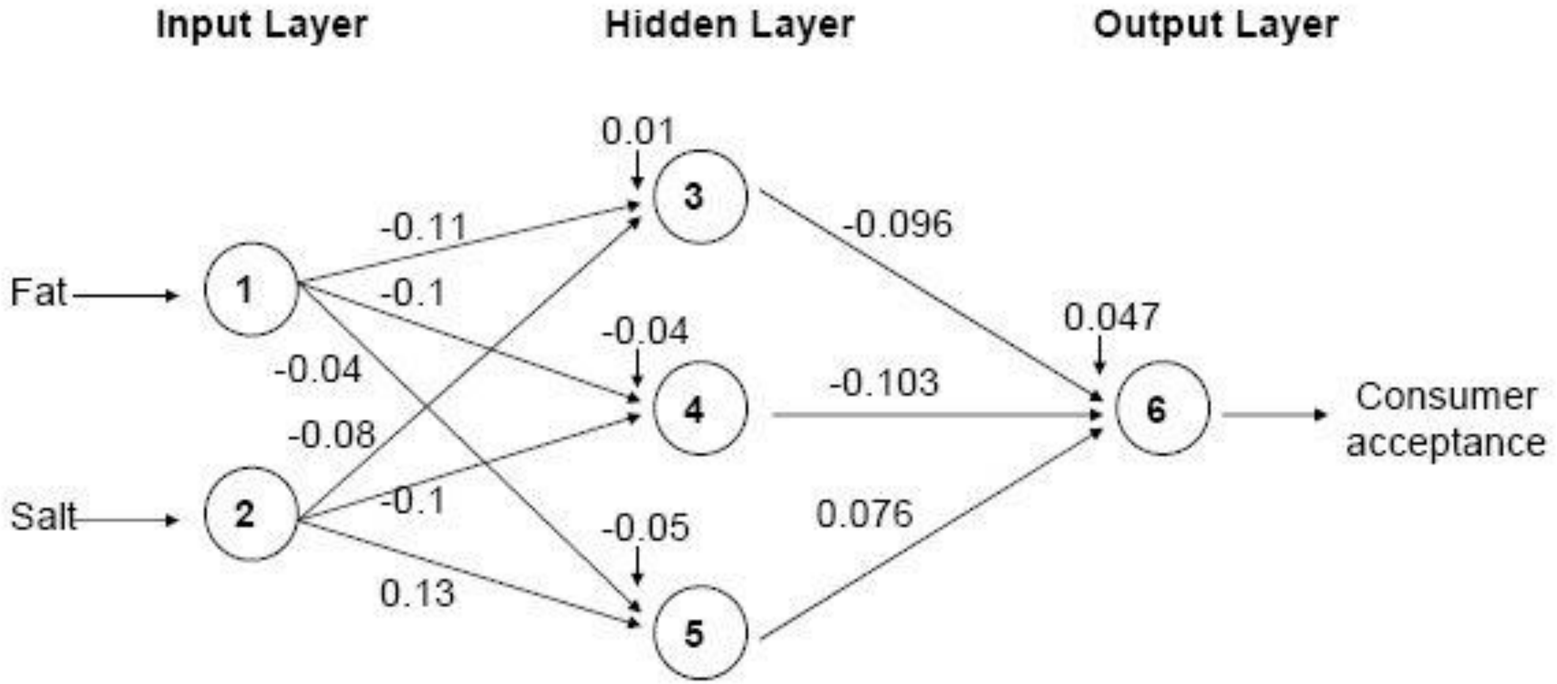
Why It Works

- Big errors lead to big changes in weights
- Small errors leave weights relatively unchanged
- Over thousands of updates, a given weight keeps changing until the error associated with that weight is negligible, at which point weights change little

Common Criteria to Stop the Updating

- When weights change very little from one iteration to the next
- When the misclassification rate reaches a required threshold
- When a limit on runs is reached

Fat/Salt Example: Final Weights



Avoiding Overfitting

With sufficient iterations, neural net can easily overfit the data

To avoid overfitting:

- Track error in validation data
- Limit iterations
- Limit complexity of network

User Inputs

Specify Network Architecture

Number of hidden layers

- Most popular – one hidden layer

Number of nodes in hidden layer(s)

- More nodes capture complexity, but increase chances of overfit

Number of output nodes

- For classification, one node per class (in binary case can also use one)
- For numerical prediction use one

Network Architecture, cont.

“Learning Rate” (*l*)

- Low values “downweight” the new information from errors at each iteration
- This slows learning, but reduces tendency to overfit to local structure

“Momentum”

- High values keep weights changing in same direction as previous iteration
- Likewise, this helps avoid overfitting to local structure, but also slows learning

Automation

- Some software automates the optimal selection of input parameters

Advantages

- Good predictive ability
- Can capture complex relationships
- No need to specify a model

Disadvantages

- Considered a “black box” prediction machine, with no insight into relationships between predictors and outcome
- No variable-selection mechanism, so you have to exercise care in selecting variables
- Heavy computational requirements if there are many variables (additional variables dramatically increase the number of weights to calculate)

Summary

- Neural networks can be used for classification and prediction
- Can capture a very flexible/complicated relationship between the outcome and a set of predictors
- The network “learns” and updates its model iteratively as more data are fed into it
- Major danger: overfitting
- Requires large amounts of data
- Good predictive performance, yet “black box” in nature

Machine Learning with Python

Session 11: Measures of Proximity, Components of Machine Learning

Arghya Ray

Introduction:

The term proximity between two objects is a function of the proximity between the corresponding attributes of the two objects. Proximity measures refer to the **Measures of Similarity and Dissimilarity**.

Similarity and Dissimilarity are important because they are used by a number of data mining techniques, such as clustering, nearest neighbour classification, and anomaly detection.

What is Similarity?

→ It is a numerical measure of the degree to which the two objects are alike.

→ Higher for pair of objects that are more alike.

→ Usually non-negative and between 0 & 1.

0 ~ No Similarity, 1 ~ Complete Similarity

What is Dissimilarity?

→ It is a numerical measure of the degree to which the two objects are different.

→ Lower for pair of objects that are more similar.

→ Range 0 to infinity.

Transformation Function

It is a function used to convert similarity to dissimilarity and vice versa, or to transform a proximity measure to fall into a particular range. For instance:

$$s' = (s - min(s)) / max(s) - min(s))$$

range

where,

s’ = new transformed proximity measure value,

s = current proximity measure value,

min(s) = minimum of proximity measure values,

max(s) = maximum of proximity measure values

This transformation function is just one example from all the available options out there.

Similarity and Dissimilarity between Simple Attributes

The proximity of objects with a number of attributes is usually defined by combining the proximities of individual attributes, so, we first discuss proximity between objects having a single attribute.

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d = \lvert x - y \rvert / (n-1)$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - d$
Interval or Ratio	$d = \lvert x - y \rvert$	$s = -d$, $s = \frac{1}{1+d}$, $s = e^{-d}$, $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

To understand it better, let us go through some examples.

Consider objects described by one **nominal** attribute. How to compare similarity of two objects like this? Nominal attributes only tell us about the distinctness of objects. Hence, in this case similarity is defined as 1 if attribute values match, and 0 otherwise and oppositely defined would be dissimilarity.

For objects with a single **ordinal** attribute, the situation is more complicated because information about order needs to be taken into account. Consider an attribute that measures the quality of a product, on the scale {poor, fair, OK, good, wonderful}. We have 3 products P1, P2, & P3 with quality as wonderful, good, & OK respectively. In order to compare **ordinal** quantities, they are mapped to successive integers. In this case, if the scale is mapped to {0, 1, 2, 3, 4} respectively. Then, dissimilarity(P1, P2) = 4−3 = 1.

For **interval or ratio** attributes, the natural measure of dissimilarity between two objects is the absolute difference of their values. For example, we might compare our current weight and our weight a year ago by saying “I am ten pounds heavier.”

Dissimilarities between Data Objects

Euclidean Distance

The Euclidean distance, d, between two points, x and y, in one, two, three, or higher- dimensional space, is given by the following formula:

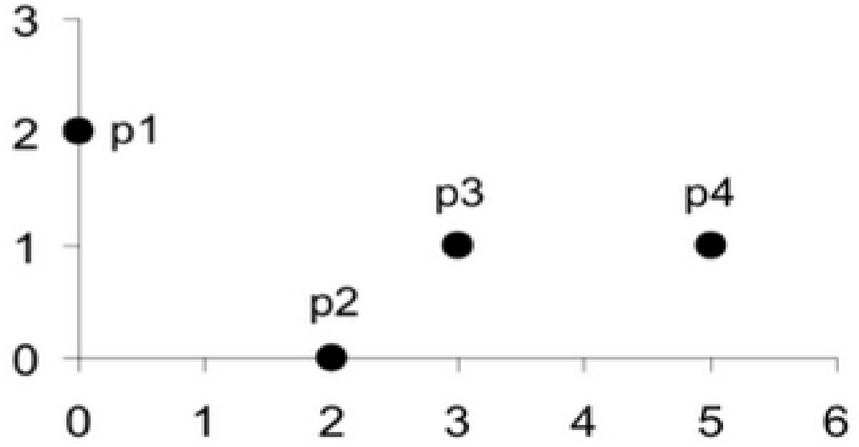
$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2},$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

where n is the number of dimensions, and *x(k)* and *y(k)* are respectively, the *kth* attributes (components) of *x* and *y*.

Dissimilarities between Data Objects

Example:



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

Dissimilarities between Data Objects

Minkowski Distance

It is the generalisation of Euclidean distance. It is given by the following formula:

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r},$$

where r is a parameter. The following are the three most common examples of Minkowski distances.

Dissimilarities between Data Objects

→ $r = 1$. City block (Manhattan, taxicab, *L1 norm*) distance. A common example is the **Hamming distance**, which is the number of bits that are different between two objects that have only binary attributes, i.e., between two binary vectors.

1 11 0001
1 1 01110

0011111 = 5

Barun
Beran

0+1+0+1+0 = 2

→ $r = 2$. Euclidean distance(*L2 norm*).

→ $r = infinity$. Supremum (*L(max), or L(infinity) norm*) distance. This is the maximum difference between any attribute of the objects. This is defined by the following formula:

$$d(\mathbf{x}, \mathbf{y}) = \lim_{r \to \infty} \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}.$$

Dissimilarities between Data Objects

Example:

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Distance Matrix

Distances, such as the Euclidean distance, have some well-known properties. If $d(x, y)$ is the distance between two points, x and y, then the following properties hold.

Positivity

a) $d(x, y) > 0$ for all x and y,

b) $d(x, y) = 0$ only if $x = y$

2. **Symmetry**

$d(x, y) = d(y, x)$ for all x and y

3. **Triangle Inequality**

$d(x, z) \leq d(x, y) + d(y, z)$ for all points x, y and z

The measures that satisfy all three properties are called **metrics.**

Similarities between Data Objects

For similarities, the triangle inequality typically does not hold, but symmetry and positivity typically do. To be explicit, if $s(x, y)$ is the similarity between points x and y, then the typical properties of similarities are the following:

$s(x, y) = 1$ only if $x = y$. $(0 \leq s \leq 1)$

$s(x, y) = s(y, x)$ for all x and y. (Symmetry)

There is no general analog of the triangle inequality for similarity measure.

Similarity Measures for Binary Data are called **similarity coefficients** and typically have values between 0 and 1. The comparison between two binary objects is done using the following four quantities:

$$f_{00} = \text{the number of attributes where } \mathbf{x} \text{ is 0 and } \mathbf{y} \text{ is 0}$$
$$f_{01} = \text{the number of attributes where } \mathbf{x} \text{ is 0 and } \mathbf{y} \text{ is 1}$$
$$f_{10} = \text{the number of attributes where } \mathbf{x} \text{ is 1 and } \mathbf{y} \text{ is 0}$$
$$f_{11} = \text{the number of attributes where } \mathbf{x} \text{ is 1 and } \mathbf{y} \text{ is 1}$$

Simple Matching Coefficient

It is defined as follows:

$$SMC = \frac{\text{number of matching attribute values}}{\text{number of attributes}} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}.$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

Jaccard Coefficient

It is defined as follows:

$$J = \frac{\text{number of matching presences}}{\text{number of attributes not involved in 00 matches}} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}.$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

An example comparing these two similarity methods:

$$\mathbf{x} = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$
$$\mathbf{y} = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)$$

$f_{01} = 2$ the number of attributes where $\mathbf{x}$ was 0 and $\mathbf{y}$ was 1
$f_{10} = 1$ the number of attributes where $\mathbf{x}$ was 1 and $\mathbf{y}$ was 0
$f_{00} = 7$ the number of attributes where $\mathbf{x}$ was 0 and $\mathbf{y}$ was 0
$f_{11} = 0$ the number of attributes where $\mathbf{x}$ was 1 and $\mathbf{y}$ was 1

$$SMC = \frac{f_{11}+f_{00}}{f_{01}+f_{10}+f_{11}+f_{00}} = \frac{0+7}{2+1+0+7} = 0.7$$

$$J = \frac{f_{11}}{f_{01}+f_{10}+f_{11}} = \frac{0}{2+1+0} = 0$$

Q. Using the given snapshot of a movie recommender system, find the Jaccard's coefficient and Matching coefficient between (a) User 1 and User 3 (b) User 2 and User 3.

	User 1	User 2	User 3
There will be blood	1	0	0
Gravity	0	1	1
X-Men	0	0	1
Inception	0	1	1
Jurassic Park	1	0	0
Avengers: End Game	1	1	1

Between User 1 & User 3:

Jaccard's Coefficient= 1/6

Matching coefficient = 1/6

Between User 2 and User 3:

Jaccard's Coefficient= 3/4

Matching coefficient = 5/6

Cosine Similarity

Documents are often represented as vectors, where each attribute represents the frequency with which a particular term(word) occurs in the document. The **cosine similarity**, is one of the most common measure of document similarity. If x and y are two document vectors, then

	cos	sin	tan
D1 →	0	1	2
D2 →	1	2	3
D3 →	2	1	1

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|},$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

where . indicates *dot product* and $||x||$ defines the length of vector x.

An example of **cosine similarity** measure is as follows:

$$\mathbf{x} = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)$$
$$\mathbf{y} = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)$$

$$\mathbf{x} \cdot \mathbf{y} = 3 * 1 + 2 * 0 + 0 * 0 + 5 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 2 * 1 + 0 * 0 + 0 * 2 = 5$$
$$\|\mathbf{x}\| = \sqrt{3 * 3 + 2 * 2 + 0 * 0 + 5 * 5 + 0 * 0 + 0 * 0 + 0 * 0 + 2 * 2 + 0 * 0 + 0 * 0} = 6.48$$
$$\|\mathbf{y}\| = \sqrt{1 * 1 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 1 * 1 + 0 * 0 + 2 * 2} = 2.24$$
$$\mathbf{cos}(\mathbf{x}, \mathbf{y}) = \mathbf{0.31}$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

Correlation

It is a measure of the linear relationship between the attributes of the objects having either binary or continuous variables. **Correlation** between two objects *x* and *y* is defined as follows:

$$\text{corr}(\mathbf{x}, \mathbf{y}) = \frac{\text{covariance}(\mathbf{x}, \mathbf{y})}{\text{standard\_deviation}(\mathbf{x}) * \text{standard\_deviation}(\mathbf{y})} = \frac{s_{xy}}{s_x \, s_y},$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

where the notations used are defined in standard as:

$$\text{covariance}(\mathbf{x}, \mathbf{y}) = s_{xy} = \frac{1}{n-1} \sum_{k=1}^{n} (x_k - \overline{x})(y_k - \overline{y})$$

$$\text{standard\_deviation}(\mathbf{x}) = s_x = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (x_k - \overline{x})^2}$$

$$\text{standard\_deviation}(\mathbf{y}) = s_y = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (y_k - \overline{y})^2}$$

$$\overline{x} = \frac{1}{n} \sum_{k=1}^{n} x_k \text{ is the mean of } \mathbf{x}$$

$$\overline{y} = \frac{1}{n} \sum_{k=1}^{n} y_k \text{ is the mean of } \mathbf{y}$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

Correlation Example: https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/

The Chi-square goodness of fit test:

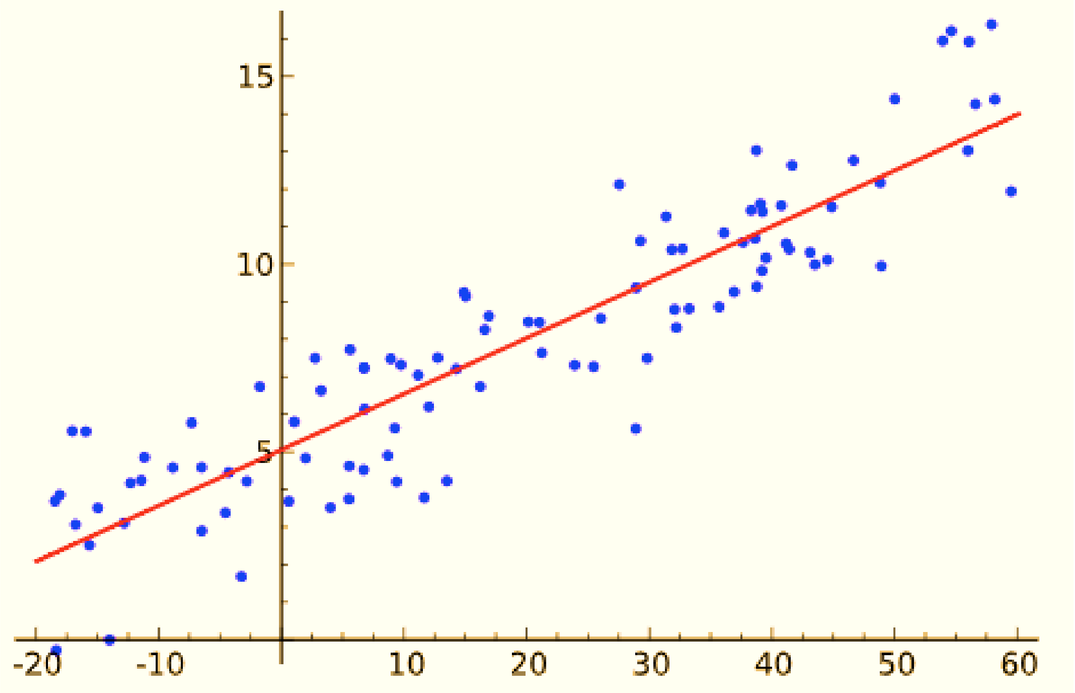
What is the goodness of fit?

A goodness-of-fit is a statistical technique. It is applied to measure "***how well the actual(observed) data points fit into a Machine Learning model***". It summarizes the divergence between actual observed data points and expected data points in context to a statistical or Machine Learning model.

Assessment of divergence between the observed data points and model-predicted data points is critical to understand, a decision made on poorly fitting models might be badly misleading. A seasoned practitioner must examine the fitment of actual and model-predicted data points.

Why do we test Goodness of fit?

Goodness-of-fit tests are statistical tests to determine whether a set of actual observed values match those predicted by the model. Goodness-of-fit tests are frequently applied in business decision making. For example, if we check linear regression function. The goodness-of-fit test here will compare the actual observed values to the predicted values.



What are the most common goodness of fit tests?

Broadly, the goodness of fit test categorization can be done based on the distribution of the predict and variable of the dataset.

- The chi-square
- Kolmogorov-Smirnov
- Anderson-Darling

The Chi-square test for a goodness-of-fit test is

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Where:

O_i= an observed count for bin*i*

E_i= an expected count for bin*i*, asserted by thenull hypothesis.

The expected frequency is calculated by:

$$E_i = \left(F(Y_u) - F(Y_l)\right) N$$

where:

F= thecumulative distribution functionfor theprobability distributionbeing tested.

Y_u= the upper limit for class*i*,

Y_l= the lower limit for class*i*, and

N= the sample size

The Chi-Square Goodness of Fit Test

Chi-square goodness of fit test is conducted when the predictand variable in the dataset is categorical. It is applied to determine whether sample data are consistent with a hypothesized distribution.

Chi-Square test can be applied when the distribution has the following characteristics:

- The sampling method is random.
- Predicted variables are categorical.
- The expected value of the number of sample observations at each level of the variable is at least 5. It requires a sufficient sample size for the chi-square approximation to be valid.

Merits of the Chi-square Test

- A distribution-free test. It can be used in any type of population distribution.
- It is widely applicable not only in social sciences but in business research as well.
- It can be easy to calculate and to conclude.
- The Chi-Square test provides an additive property. This allows the researcher to add the result of independence to related samples.
- This test is based on the observed frequency and not on parameters like mean, and standard deviation.

Until now we have defined and understood both similarity and dissimilarity measures amongst data objects. Now, let's discuss the issues faced in proximity calculations.

Issues in Proximity Calculation

- how to handle the case in which attributes have different scales and/or are correlated,
- how to calculate proximity between objects that are composed of different types of attributes, e.g., quantitative and qualitative, and
- how to handle proximity calculation when attributes have different weights i.e., when not all attributes contribute equally to the proximity of objects.

Selecting the Right Proximity Measure

The following are a few general observations that may be helpful. First, ***the type of proximity measure should fit the type of data***. For many types of dense, continuous data, metric distance measures such as Euclidean distance are often used. Proximity between continuous attributes is most often expressed in terms of differences, and distance measures provide a well-defined way of combining these differences into an overall proximity measure.

For sparse data, which often consists of asymmetric attributes, we typically employ similarity measures that ignore 0–0 matches. Conceptually, this reflects the fact that, for a pair of complex objects, similarity depends on the number of characteristics they both share, rather than the number of characteristics they both lack. For such type of data, Cosine Similarity or Jaccard Coefficient can be used.

Main Components of Machine Learning Algorithm:

1) Feature Extraction + Domain knowledge

First and foremost we really need to understand what type of data we are dealing with and what eventually we want to get out of it. Essentially we need to understand how and what features need to be extracted from the data. For instance assume we want to build a software that distinguishes between male and female names. All the names in text can be thought of as our raw data while our features could be number of vowels in the name, length, first & last character, etc of the name.

2) Feature Selection

In many scenarios we end up with a lot of features at our disposal. We might want to select a subset of those based on the resources and computation power we have. In this step we select a few of those influential features and separate them from the not-so-influential features. There are many ways to do this, information gain, gain ratio, correlation etc.

3) Choice of Algorithm

There are wide range of algorithms from which we can choose based on whether we are trying to do prediction, classification or clustering. We can also choose between linear and non-linear algorithms. Naive Bayes, Support Vector Machines, Decision Trees, k-Means Clustering are some common algorithms used.

4) Training

In this step we tune our algorithm based on the data we already have. This data is called training set as it is used to train our algorithm. This is the part where our machine or software learn and improve with experience.

5) Choice of Metrics/Evaluation Criteria

Here we decide our evaluation criteria for our algorithm. Essentially we come up with metrics to evaluate our results. Commonly used measures of performance are precision, recall, f1-measure, robustness, specificity-sensitivity, error rate etc.

6) Testing

Lastly, we test how our machine learning algorithm performs on an unseen set of test cases. One way to do this, is to partition the data into training and testing set. The training set is used in step 4 while the test set is then used in this step. Techniques such as cross-validation and leave-one-out can be used to deal with scenarios where we do not have enough data.

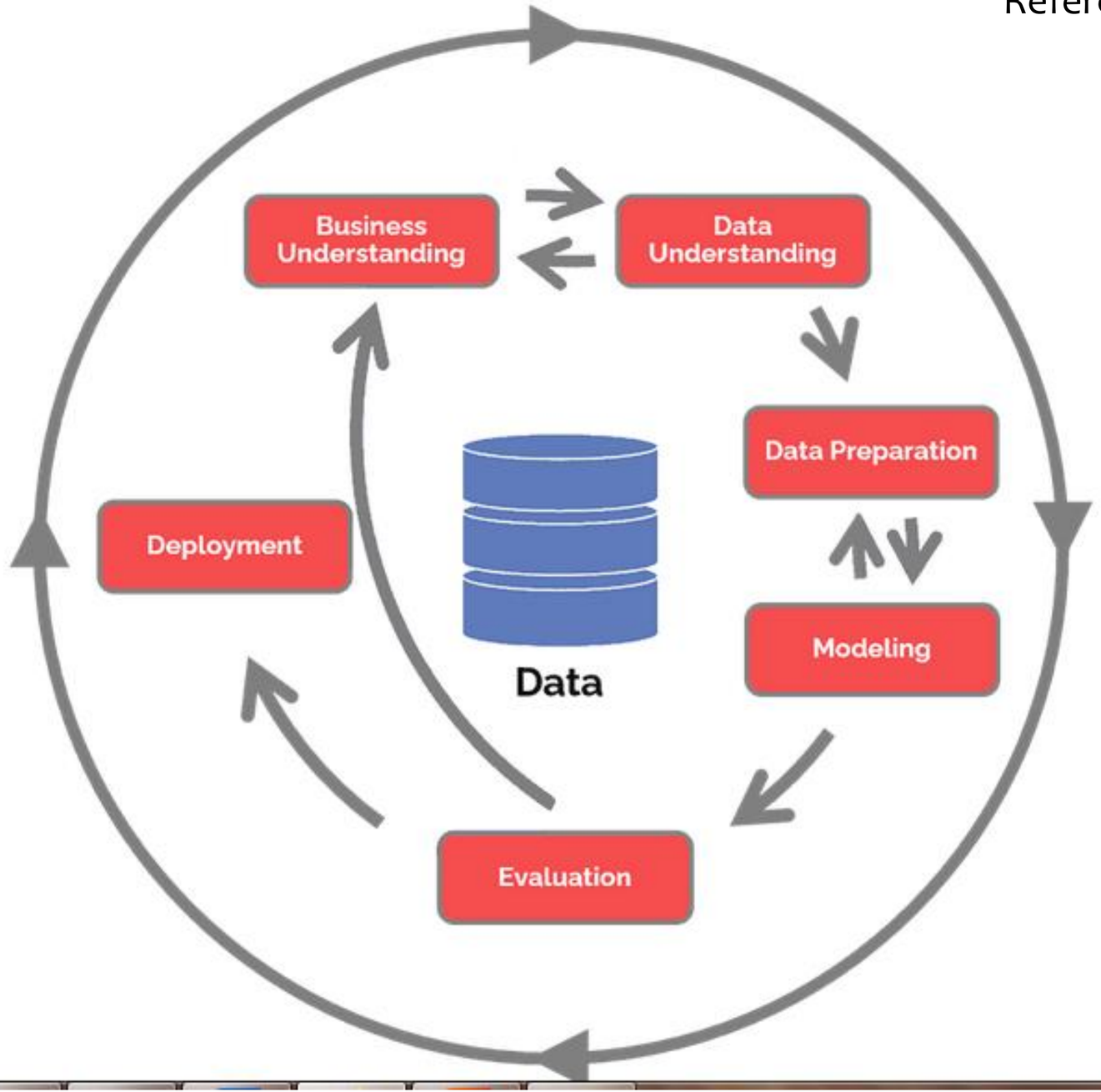
(Reference: https://www.linkedin.com/pulse/20140822073217-180198720-6-components-of-a-machine-learning-algorithm)

Main steps involved for End-to-End Machine Learning Project: (Chapter 2 of textbook)

1. Look at the big picture
2. Get the data
3. Discover and visualize the data to gain insights
4. Prepare the data for Machine Learning algorithms
5. Select a model and train it
6. Fine tune your model
7. Present your solution
8. Launch, monitor and maintain your system

CRISP DM (The **CR**oss **I**ndustry **S**tandard **P**rocess for **D**ata **M**ining)

Reference: https://www.datascience-pm.com/crisp-dm-2/



The content of the slides are prepared from different textbooks.

References:

Proximity Measures:

- https://towardsdatascience.com/measures-of-proximity-in-data-mining-machine-learning-e9baaed1aafb

Chi-Square Goodness of Fit readings:

- https://www.mygreatlearning.com/blog/understanding-goodness-of-fit-test/
- https://machinelearningmastery.com/chi-squared-test-for-machine-learning/
- https://towardsdatascience.com/machine-learning-chi-square-test-in-evaluating-predictions-486404dd5bc
- https://towardsdatascience.com/chi-square-test-for-feature-selection-in-machine-learning-206b1f0b8223
- https://www.analyticsvidhya.com/blog/2019/11/what-is-chi-square-test-how-it-works/ (stepwise calculation)
- https://medium.com/wenyi-yan/a-simple-explanation-to-understand-chi-square-test-1814fa261499 (step wise calculation simple)

Correlation:

https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

Extra Read:

https://towardsdatascience.com/types-of-data-sets-in-data-science-data-mining-machine-learning-eb47c80af7a


Thank you..

Machine Learning with Python

Session 12: Cluster Analysis

Arghya Ray

Identifying Similarities in Data

- Large amount of information are constantly being generated, organized, analyzed and stored.
- Identifying important patterns, associations and groupings of similar data can be helpful for customers as well as organizations.
- **Data Clustering** can help us make sense of the huge amount of data by discovering hidden groupings of similar items.
- Clustering can help in analysis of online social networks.
- Clustering can help to distinguish between different items. E.g. Fresh vegetables are more similar to each other than frozen items.
- Clustering is useful in ***market segmentation*** by partitioning the target market data into groups such as customer who share the same interests or those with common needs. Identifying clusters of similar items can help develop a marketing strategy that addresses the needs of specific clusters.
- Data Clustering can help ***to identify, learn, or predict the nature of new data items***– especially how new data can be linked with making predictions. For e.g., in pattern recognition analyzing patterns in the data (such as buying patterns in particular regions or age groups) can help to develop predictive analytics to predict the nature of future data items that can fit well with established patterns.
- Clustering can help in ***dividing the e-mail dataset into spam and non-spam messages***.
- Data Clustering is also helpful ***in image segmentation*** for analyzing the image more easily.
- Data Clustering can help ***in information retrieval from a collection of data***, mainly documents (using tf-idf concepts).
- On the other hand, finding important **association rules in a dataset** of customer transactions helps a company to maximize revenue by deciding which products should be on sale, how to position products in the store's aisles, and how and when to offer promotional pricing.

Types of Cluster Analysis Methods:

- **Partitional Methods**: Partitional methods obtain a single level partition of objects. These methods are usually based on a greedy heuristics that are used to obtain a local optimum solution. Given *n* objects, these methods make k<=n clusters.
  - ***K-means***: Each of the K-clusters is represented by the mean of the objects inside each cluster.
  - ***Density-Based***: It is based on the assumption that clusters have high density collection of data of arbitrary shape that are separated by a large space of low density data (which is assumed to be the noise).
  - ***Expectation-Maximization***: The EM method assigns objects to different clusters with certain probabilities in an attempt to maximize the expectation (or likelihood) of assignment.
- **Hierarchical Methods**: Hierarchical methods obtain a nested partition of the objects resulting in a tree of clusters.
  - ***Agglomerative***: Start with each object in an individual cluster and then try to merge similar clusters into larger and larger clusters.
  - ***Divisive***: Start with one cluster and then split into smaller and smaller clusters.
- **Grid Based method**: The object space rather than the data is divided into a grid. Grid partitioning is based on characteristics of the data and such methods can deal with non-parametric data more easily.
- **Model based method**: A model is assumed based on a probability distribution. Essentially the algorithm tries to build clusters with a high level of similarity with them and a low level of similarity between them. It tries to minimise the squared-error function.

Requirements of Clustering in Data Mining

The following points throw light on why clustering is required in data mining –

Scalability – We need highly scalable clustering algorithms to deal with large databases.

Ability to deal with different kinds of attributes – Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) data, categorical, and binary data.

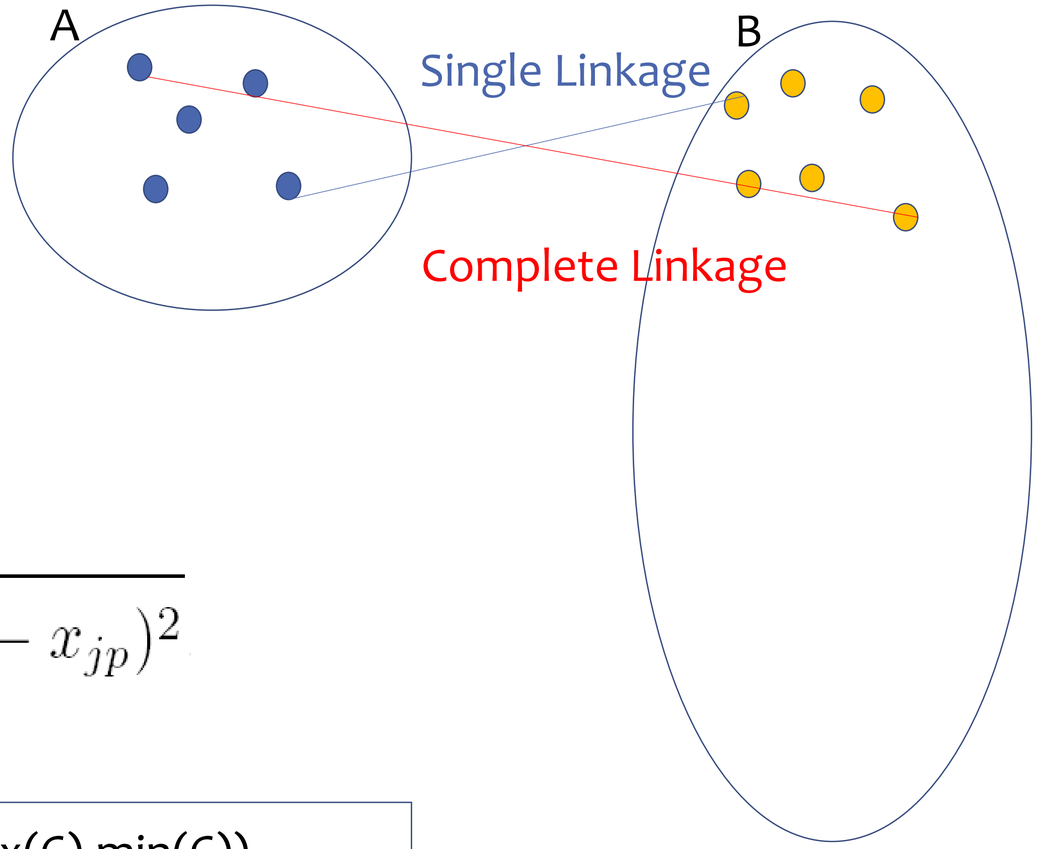
Discovery of clusters with attribute shape – The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small sizes.

High dimensionality – The clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.

Ability to deal with noisy data – Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.

Interpretability – The clustering results should be interpretable, comprehensible, and usable.

Some important concepts:



- **Measuring Distance**
  - Between records: Distance between each record in a cluster.
  - Between clusters: Distance between each cluster.
- **Distance Between Two Records:** Euclidian distance is most popular.

$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

- **Normalizing:**

(x-min(C))/(max(C)-min(C))

  - ***Problem***: Raw distance measures are highly influenced by scale of measurements
  - ***Solution***: Normalize (standardize) the data first.
  - Subtract mean, divide by std. deviation. (Also called z-scores).
  - Example: For 22 utilities, Avg. sales = 8,914; Std. dev. = 3,550. Hence, Normalized score: (9,077-8,914)/3,550 = 0.046

Measuring Distance Between Clusters:

- **Single Linkage**
  - Minimum Distance (Cluster A to Cluster B)
  - Distance between two clusters is the distance between the pair of records Ai and Bj that are closest.
- **Complete Linkage**
  - Maximum Distance (Cluster A to Cluster B)
  - Distance between two clusters is the distance between the pair of records Ai and Bj that are farthest from each other
- **Average Linkage**
  - Distance between two clusters is the average of all possible pair-wise distances
- **Centroid**
  - Distance between two clusters is the distance between the two cluster centroids.
  - Centroid is the vector of variable averages for all records in a cluster

K-Means Clustering:

- The K-means method may be described as follows:
  1. Select the number of clusters. Let this be ***k***.
  2. Pick k seeds as centroids of the ***k*** clusters. The seeds may be picked randomly unless the user has some insight about the data.
  3. Compute the Euclidean distance of each object in the dataset from each of the centroids.
  4. Allocate each object to the cluster it is nearest to based on the distances computed in the previous step.
  5. Compute the centroids of the clusters by computing the means of the attribute values of the objects in each cluster.
  6. Check if the stopping criteria has been met. If yes, stop. If not, go to step 3.
- The ***k-means method*** uses the Euclidean distance method, which appears to work well with compact clusters.
- If the Manhattan distance is used the method is called ***k-median method***. This method may be less sensitive to outliers.
- K-means Algorithm: Choosing k and Initial Partitioning

  Manhattan Distance-> $|x1-x2| + |y1-y2|$
  - Choose k based on the how results will be used. E.g., "How many market segments do we want?"
  - Also experiment with slightly different k's.
  - Initial partition into clusters can be random, or based on domain knowledge. If random partition, repeat the process with different random partitions
- For clustering to be effective all attributes should be converted to a similar scale unless you want to give more weight to some attributes that are relatively large in scale.

1 6 3 7 4 9

Step 1: K= 2

Step 2: C1: 1 6 3 C2: 7 4 9
Mean 3.333 6.667

Step 3: **C1:** **1** 6 3
M1: 2.3333 2.667 0.3333
M2: 5.667 0.667 3.667

C2: **7** **4** **9**
M2: 1.667 2.667 2.333
M1: 4.337 0.7 5.7

Step 4: 1 → C1; 6→C2; 3→C 1 ; 7→C 2 ; 4→C 1 ; 9→C2

Step 5: C1: 1 3 4 C2: 6 7 9
Mean 2.667 7.333

Step 6: C1: 1 3 4 C2: 6 7 9
M1: 1.667 0.333 1.33 M1: 3.333 4.333 6.333
M2: 6.333 4.333 3.333 M2: 1.333 0.333 1.67

Step 7: 1→ C1; 3→C1; 4→ C1 6→ C2; 7→ C2; 9→C2

Q1. Use k-mean clustering to divide the following set of numbers into two clusters.

1 2 3 4 5 6 7 8 9 10

Q2. Use k-median method to form clusters of students based on the data given below:

Student	Age	Mark1	Mark2	Mark3
S1	18	73	75	57
S2	18	79	85	75
S3	23	70	70	52
S4	20	55	55	55
S5	22	85	86	87
S6	19	91	90	89
S7	20	70	65	60
S8	21	53	56	59
S9	19	82	82	60
S10	47	75	76	77

K=3

Steps 1 and 2: Let the three seeds be the first three students as shown.

Student	Age	Mark1	Mark2	Mark3
S1	18	73	75	57
S2	18	79	85	75
S3	23	70	70	52

Step 3 and 4: Compute the distances using the four attributes and using the sum of absolute differences (k-Median method)

	Age	Mark1	Mark2	Mark3	Distance from Clusters			Allocation to the nearest Cluster
C1	18	73	75	57	From C1	From C2	From C3	
C2	18	79	85	75				
C3	23	70	70	52				
S1	18	73	75	57	0	34	18	C1
S2	18	79	85	75	34	0	52	C2
S3	23	70	70	52	18	52	0	C3
S4	20	55	55	55	42	76	36	C3
S5	22	85	86	87	57	23	67	C2
S6	19	91	90	89	66	32	82	C2
S7	20	70	65	60	18	46	16	C3
S8	21	53	56	59	44	74	40	C3
S9	19	82	82	60	20	22	36	C1
S10	47	75	76	77	52	44	60	C2

Step 5: The new cluster means of clusters are given in the table below:

Manhattan Distance-> $|x1-x2| + |y1-y2|+|z1-z2|+|w1-w2|$

Student	Age	Mark1	Mark2	Mark3
C1	18.5	77.5	78.5	58.5
C2	26.5	82.5	84.3	82.0
C3	21	61.5	61.5	65.5

Step 3 and 4: Using this new cluster means compute the distances of each object to each of the means and allocate to nearest cluster.

	Age	Mark1	Mark2	Mark3	Distance from Clusters			Allocation to the nearest Cluster
C1	18.5	77.5	78.5	58.5	From C1	From C2	From C3	
C2	26.5	82.5	84.3	82.0				
C3	21	61.5	61.5	65.5				
S1	18	73	75	57	10	52.3	28	C1
S2	18	79	85	75	25	19.8	62	C2
S3	23	70	70	52	27	60.3	23	C3
S4	20	55	55	55	51	90.3	16	C3
S5	22	85	86	87	47	13.8	79	C2
S6	19	91	90	89	56	28.8	92	C2
S7	20	70	65	60	24	60.3	16	C3
S8	21	53	56	59	50	86.3	17	C3
S9	19	82	82	60	10	32.3	46	C1
S10	47	75	76	77	52	41.3	74	C2

Step 6: The clusters have not changed and hence we can stop.
Therefore, **Cluster 1:** S1, S9. **Cluster 2:** S2, S5, S6, S10. **Cluster 3:** S3, S4, S7, S8.

Cluster	C1	C2	C3
C1	5.9	26.5	23.3
C2	29.5	14.3	42.6
C3	23.9	41.0	13.7

Within cluster and between cluster distances

The content of the slides are prepared from different textbooks.

References:

- Data Mining and Predictive Analytics, By Daniel T. Larose. Copyright 2015 John Wiley & Sons, Inc.
- Predictive Analytics for Dummies, By Anasse Bari, Mohamed Chaouchi, & Tommy Jung, Copyright 2016, John Wiley & Sons, Inc.
- Introduction to Data Mining with Case Studies, By G.K. Gupta. Copyright 2014 by PHI Learning Private Limited.


Thank you..

Machine Learning with Python

Session 12: Cluster Analysis

Arghya Ray

Rational for Measuring Cluster Goodness

- What are the optimal number of cluster to identify?
- How do I measure weather one set of clusters is preferable to another?
- The **silhouette** method and the **pseudo-F statistic** will help us address these questions by measuring cluster goodness.

Concepts Measures Should Address

- Cluster separation represents how distant the clusters are from each other
- Cluster cohesion refers to how tightly related the records within the individual clusters are
- Good measures should incorporate both as do the silhouette and pseudo-F statistic
- However, the sum of squares error (SSE) only accounts for cluster cohesion and is monotonically decreasing with increasing numbers of clusters.

Measuring Cluster Goodness: **The Silhouette Method**

For each data value *i* the silhouette is used to gauge how good the cluster assignment is for that point:

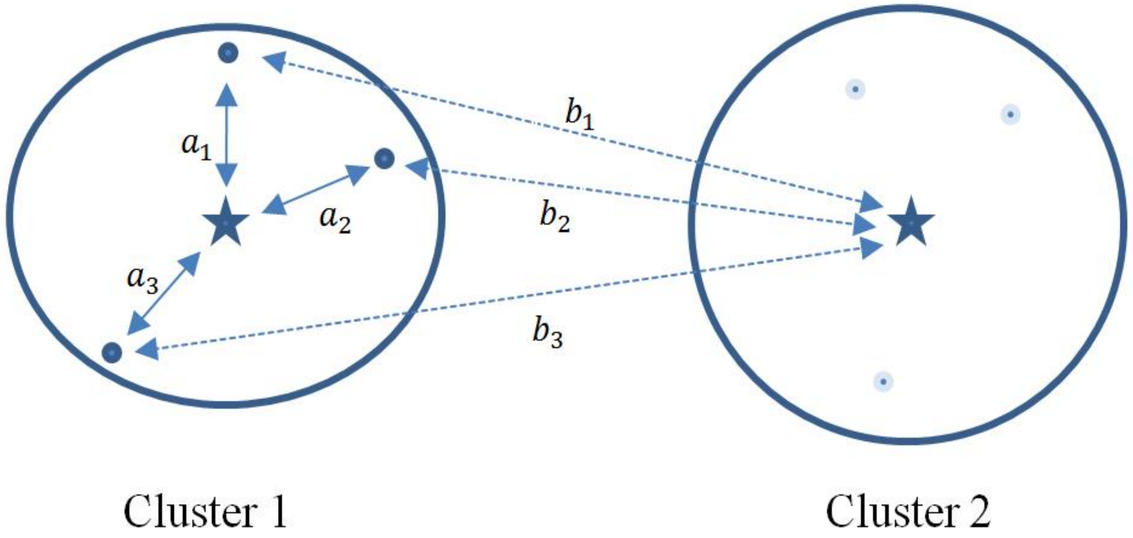
$$\boldsymbol{Silhouette_i = s_i = \frac{b_i - a_i}{\max(b_i, a_i)}}$$

where a_i is the distance between the data value and its cluster center and represents *cohesion*

and b_i is the distance between the data value and the next closest cluster center and represents *separation*

Silhouette Accounts for Separation & Cohesion

Each data value in Cluster 1 has its values of ai and bi represented by solid and dotted lines, respectively



bi > ai for each data value, thus each data value's silhouette is positive, indicating the data are not misclassified

Measuring Cluster Goodness: **The Silhouette Method contd…**

- A positive value indicates that the assignment is good, with higher values better than lower values.
- A value close to zero is considered to be weak since the observation could have been assigned to the next cluster with little negative consequence.
- A negative value is considered to be misclassified since assignment to the next closest cluster would have been better.

The Average Silhouette Value

The average silhouette value over all records yields a measure of how well the cluster solution fits. A thumbnail interpretation, meant as a guide only:

- 0.5 or better provides good evidence of the reality of the clusters in the data
- 0.25 – 0.5 provides some evidence of the reality of the clusters in the data.
- Less than 0.25 provides scant evidence of cluster reality

Silhouette Example (cont.)

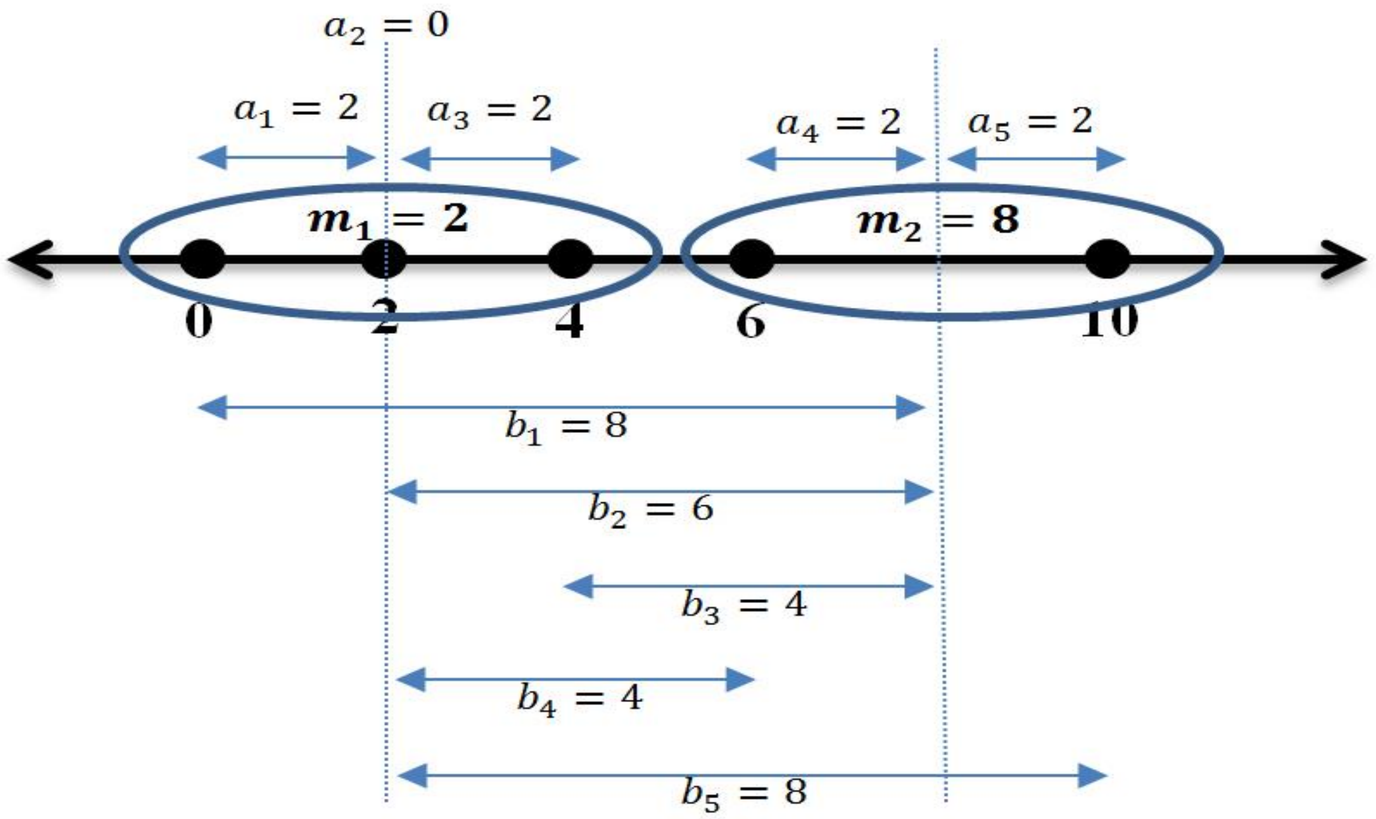
- Apply *k*-means clustering to the following data set:

$x_1 = 0 \quad x_2 = 2 \quad x_3 = 4 \quad x_4 = 6 \quad x_5 = 10$

- The first three data values are assigned to Cluster 1 and the last two to Cluster 2
- Center for Cluster 1 is $m_1 = 2$ and for Cluster 2 is $m_2 = 8$
- Values for a_i are distance between x_i and its cluster center; values for b_i are distance between x_i and the other cluster center

Silhouette Example (cont.)

Distances between the data values and cluster centers:



Silhouette Example (cont.)

Calculations for individual data values:

x_i	a_i	b_i	$Max(a_i, b_i)$	$\text{Silhouette}_i = s_i = \frac{b_i - a_i}{\text{Max}(b_i, a_i)}$
0	2	8	8	$\frac{8-2}{8}$ =0.75
2	0	6	6	$\frac{6-0}{6}$ =1.00
4	2	4	4	$\frac{4-2}{4}$ =0.50
6	2	4	4	$\frac{4-2}{4}$ =0.50
10	2	8	8	$\frac{8-2}{8}$ =0.75
				Mean Silhouette = 0.7

The pseudo-*F* Statistic

Let:

k be number of clusters

$\sum n_i = N$ betotal sample size

x_{ij} refer to the j^{th} data value in the i^{th} cluster

m_i refer to cluster center (centroid) of the i^{th} cluster

M represent the grand mean of all the data

and $Distance(a, b) = \sqrt{\sum (a_i - b_i)^2}$

The pseudo-*F* Statistic (cont.)

Then the *sum of squares between* the clusters is:

$$SSB = \sum_{i=1}^{k} n_i \cdot Distance^2(m_i, M)$$

And the *sum of squares error*, or the *sum of squares within* the clusters is:

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n_j} Distance^2(x_{ij}, m_i)$$

And the *pseudo-F statistic* is:

$$F = \frac{MSB}{MSE} = \frac{SSB/k-1}{SSE/N-k}$$

The pseudo-*F* Statistic (cont.)

- The hypotheses being tested are:

$$H_0: \text{There are no clusters in the data.}$$
$$H_a: \text{There are } k \text{ clusters in the data.}$$

- Reject H_o for sufficiently small p-value where:

$$p\text{-}value = P(F_{k-1,n-k}) > \text{Pseudo F value})$$

- The pseudo-F statistic rejects the null hypothesis too easily.

The pseudo-*F* Statistic (cont.)

The pseudo-F statistic should not be used to determine the presence of clusters but can be used to select the optimal number of clusters as follows:

1. Use a clustering algorithm to develop a clustering solution for a variety of values of k.

2. Calculate the pseudo-F statistic and p-value for each candidate, and select the candidate with the smallest p-value as the best clustering solution.

Pseudo-*F* Statistic Example

- Apply *k*-means clustering to the following data set:

$x_1 = 0 \quad x_2 = 2 \quad x_3 = 4 \quad x_4 = 6 \quad x_5 = 10$

- The first three data values are assigned to Cluster 1 and the last two to Cluster 2
- Center for Cluster 1 is $m_1 = 2$ and for Cluster 2 is $m_2 = 8$
- $n_1 = 3$ and $n_2 = 2$ data values, and $N = 5$, the grand mean is $M = 4.4$. And, because we are in one dimension, $Distance(m_i, M) = |m_i - M|$

Pseudo-*F* Statistic Example (cont.)

Then

$$SSB = \sum_{i=1}^{k} n_i \cdot Distance^2(m_i, M)$$
$$= 3 \cdot (2 - 4.4)^2 + 2 \cdot (8 - 4.4)^2 = 43.2$$

And

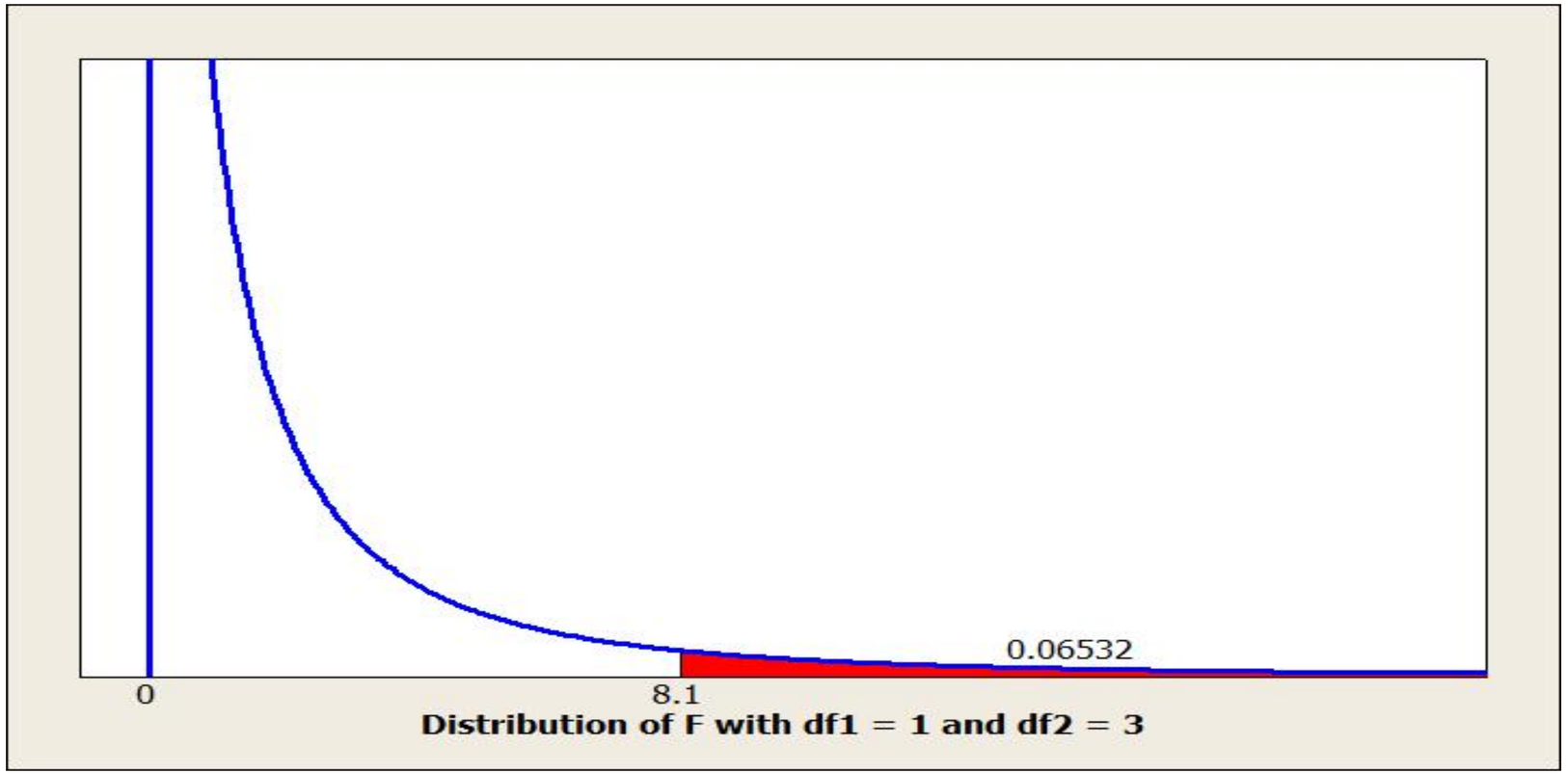
$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n_j} Distance^2(x_{ij}, m_i)$$
$$= (0 - 2)^2 + (2 - 2)^2 + (4 - 2)^2 + (6 - 8)^2 + (10 - 8)^2 = 16$$

And

$$F = \frac{MSB}{MSE} = \frac{SSB/k - 1}{SSE/N - k} = \frac{43.2/1}{16/3} = \frac{43.2}{5.33} = 8.1$$

Pseudo-*F* Statistic Example (cont.)

Distribution of the F statistic shows that p-value of 0.06532 does not indicate strong evidence of clusters:



Cluster Validation

- As with any other data mining modeling technique, cluster analysis should be subject to cross-validation to ensure the clusters are real
- A simple graphical and statistical approach with the goal of confirming the clusters found in the test data match those in the training data is:

1. Apply cluster analysis to training data
2. Apply cluster analysis to test data
3. Use graphics and statistics to confirm the clusters in the training data match those in the test data

The content of the slides are prepared from different textbooks.

References:

- Data Mining and Predictive Analytics, By Daniel T. Larose. Copyright 2015 John Wiley & Sons, Inc.
- Predictive Analytics for Dummies, By Anasse Bari, Mohamed Chaouchi, & Tommy Jung, Copyright 2016, John Wiley & Sons, Inc.
- Introduction to Data Mining with Case Studies, By G.K. Gupta. Copyright 2014 by PHI Learning Private Limited.


Thank you..

Machine Learning with Python

Session 14: Agglomerative Clustering

Arghya Ray

Measuring Distance Between Clusters:

- **Single Linkage**
  - Minimum Distance (Cluster A to Cluster B)
  - Distance between two clusters is the distance between the pair of records Ai and Bj that are closest.
- **Complete Linkage**
  - Maximum Distance (Cluster A to Cluster B)
  - Distance between two clusters is the distance between the pair of records Ai and Bj that are farthest from each other
- **Average Linkage**
  - Distance between two clusters is the average of all possible pair-wise distances
- **Centroid**
  - Distance between two clusters is the distance between the two cluster centroids.
  - Centroid is the vector of variable averages for all records in a cluster

$(x1, y1, z1)$ $(x2, y2, z2)$ $(x3, y3, z3)$

$C1 = ((x1+x2+x3)/3, (y1+y2+y3)/3, (z1+z2+z3)/3)$

Q. Consider the following three clusters, each with four members:

Cluster 1: {(1,5), (2,4), (3,3), (2,1)} Centroid- ((1+2+3+2)/4, (5+4+3+1)/4) (2,3.25)

Cluster 2: {(5,4), (6,6), (7,5), (8,8)}

Cluster 3: {(4,1), (3,0), (5,1), (6,2)}

Distance Between	Single Link	Complete-Link	Centroid	Average-Link
Cluster 1 & 2	2.24	9.22	5.15	5.43
Cluster 2 & 3	2.24	9.43	5.15	5.38
Cluster 3 & 1	1.41	5.83	3.36	3.76

- Whichever distance algorithm is applied, two nearby clusters can be merged if an agglomerative approach is being used.
- It has been reported that the ***complete link algorithm*** generally produces compact and more useful clusters.
- The ***single link algorithm*** tends to suffer from chaining effects and elongated clusters in some situations. However, the single link algorithm is found to be effective in some applications.
- Both the complete link algorithm and single link algorithm can suffer from the presence of outliers.

Agglomerative Method:

The basic idea of the agglomerative method is to start out with n-clusters for ‘n’ data points and keep on combining points.

Steps involved:

1. Allocate each point to a cluster of its own. Thus we start with *n* clusters for *n* objects.
2. Create a distance matrix by computing distances between all pairs of clusters using one of the distance measuring methods (e.g. single link metric or complete link metric). Sort these distances in ascending order.
3. Find the two clusters that have the smallest distance between them.
4. Remove the pair of objects and merge them. When you are merging, take the average value of the two cluster distances.
5. If there is only one cluster left, then stop.
6. Compute all distances from the new cluster and update the distance matrix after the merger and go to step 3.

Use agglomerative clustering method for clustering the data (use centroid method for calculating distance between clusters).

Student	Age	Mark1	Mark2	Mark3
S1	18	73	75	57
S2	18	79	85	75
S3	23	70	70	52
S4	20	55	55	55
S5	22	85	86	87
S6	19	91	90	89
S7	20	70	65	60
S8	21	53	56	59
S9	19	82	82	60
S10	47	75	76	77

S1→ (18,73,75,57)
S2→ (18,79,85,75)

S1 and S2→ 34
S2 and S3→ 52
S1 and S3→ 18
S2 and S4 → 76

Step 1 and 2: Allocate each point to a cluster and compute the distance matrix using the centroid method. The distance matrix is symmetric.

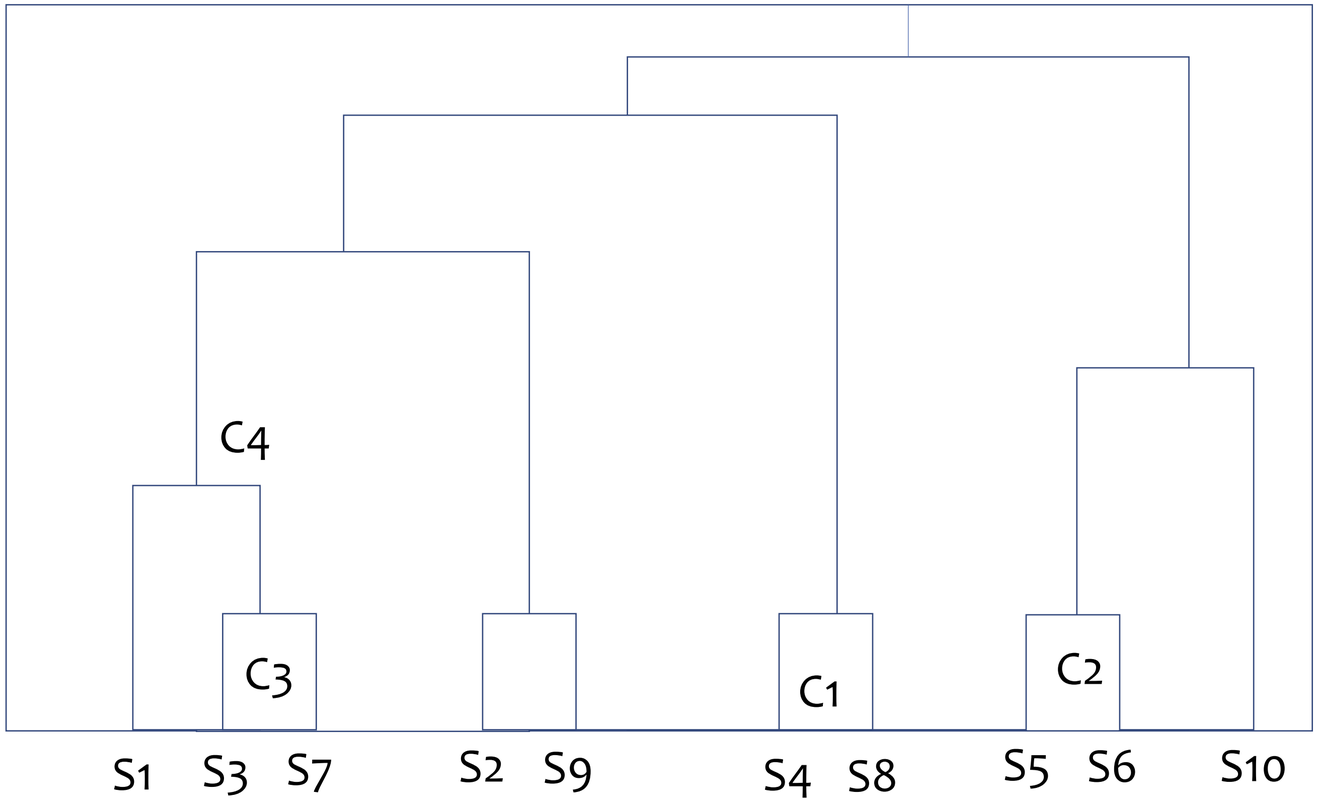
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	0									
S2	34	0								
S3	18	52	0							
S4	42	76	36	0						
S5	57	23	67	95	0					
S6	66	32	82	106	15	0				
S7	18	46	16	30	65	76	0			
S8	44	74	40	8	91	104	28	0		
S9	20	22	36	60	37	46	30	58	0	
S10	52	44	60	90	55	70	60	86	58	0

Step 3 and 4: The smallest distance is 8 between objects S4 and S8. We combine this to cluster (C1) and put it where S4 was.

	S1	S2	S3	C1	S5	S6	S7	S9	S10
S1	0								
S2	34	0							
S3	18	52	0						
C1	41	75	38	0					
S5	57	23	67	95	0				
S6	66	32	82	106	15	0			
S7	18	46	16	29	65	76	0		
S9	20	22	36	59	37	46	30	0	
S10	52	44	60	88	55	70	60	58	0

Step 5 and 6: The smallest distance is now 15 between objects S5 and S6. We combine this to cluster (C2) and put it where S5 was.

	S1	S2	S3	C1	C2	S7	S9	S10
S1	0							
S2	34	0						
S3	18	52	0					
C1	41	75	38	0				
C2	61.5	27.5	74.5	97.5	0			
S7	18	46	16	29	69.5	0		
S9	20	22	36	59	41.5	30	0	
S10	52	44	60	88	62.5	60	58	0



	S1	S2	S3	C1	C2	S9	S10
S1	0						
S2	34	0					
C3	15	49	0				
C1	41	75	30	0			
C2	61.5	27.5	71.5	97.5	0		
S9	20	22	33	59	41.5	0	
S10	52	44	60	88	62.5	58	0

Divisive Hierarchical Method:

The basic idea of the divisive method is that it starts with the whole dataset as one cluster and then proceeds to recursively divide the cluster into two sub-clusters and continues until each cluster has only one object or some other stopping criterion has been reached. There are two types of divisive methods:

- ***Monothetic***: It splits a cluster using only one attribute at a time.
- ***Polythetic***: It splits a cluster using all attributes together.

Steps involved in a polythetic divisive method:

1. Decide a method of measuring the distance between two objects. Also decide a threshold distance.
2. Create a distance matrix by computing distance between all pairs of objects within the cluster. Sort these distances in ascending order.
3. Find the two objects that have the largest distance between them. They are most dissimilar.
4. If the distance between the two objects is smaller than the pre-specified threshold and there is no other cluster that needs to be divided then stop, otherwise continue.
5. Use the pair of objects as seeds of a K-means method to create two new clusters
6. If there is only one object in each cluster, then stop otherwise continue with Step 2.

Use divisive clustering method for clustering the data (use centroid method for calculating distance between clusters).

Student	Age	Mark1	Mark2	Mark3
S1	18	73	75	57
S2	18	79	85	75
S3	23	70	70	52
S4	20	55	55	55
S5	22	85	86	87
S6	19	91	90	89
S7	20	70	65	60
S8	21	53	56	59
S9	19	82	82	60
S10	47	75	76	77

Step 1 and 2: Allocate each point to a cluster and compute the distance matrix using the centroid method. The distance matrix is symmetric.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	0									
S2	34	0								
S3	18	52	0							
S4	42	76	36	0						
S5	57	23	67	95	0					
S6	66	32	82	106	15	0				
S7	18	46	16	30	65	76	0			
S8	44	74	40	8	91	104	28	0		
S9	20	22	36	60	37	46	30	115	0	
S10	52	44	60	90	55	70	60	98	99	0

The largest distance is 115 between the objects S8 and S9. They becomes the seed for to new clusters. K-Means is used to split the group into two clusters.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S8	44	74	40	8	91	104	28	0	115	98
S9	20	22	36	60	37	46	30	115	0	99

Cluster C1: S4, S7, S8, S10
Cluster C2: S1, S2, S3, S5, S6, S6, S9

Cluster C1: S4, S7, S8, S10

Cluster C2: S1, S2, S3, S5, S6, S6, S9

If the stopping criteria is not met, we can follow the previous steps and divide these two clusters again one by one.

For Cluster C2:

	S1	S2	S3	S5	S6	S9
S1	0					
S2	34	0				
S3	18	52	0			
S5	57	23	67	0		
S6	66	32	82	15	0	
S9	20	22	36	37	46	0

The largest distance is 82 between the objects S3 and S6. They becomes the seed for to new clusters.

For Cluster C1:

	S4	S7	S8	S10
S4	0			
S7	30	0		
S8	8	28	0	
S10	90	60	98	0

The largest distance is 98 between the objects S8 and S10. They becomes the seed for to new clusters.

This continues until one of the stopping criteria is met.

The content of the slides are prepared from different textbooks.

References:

- Data Mining and Predictive Analytics, By Daniel T. Larose. Copyright 2015 John Wiley & Sons, Inc.
- Predictive Analytics for Dummies, By Anasse Bari, Mohamed Chaouchi, & Tommy Jung, Copyright 2016, John Wiley & Sons, Inc.
- Introduction to Data Mining with Case Studies, By G.K. Gupta. Copyright 2014 by PHI Learning Private Limited.


Thank you..