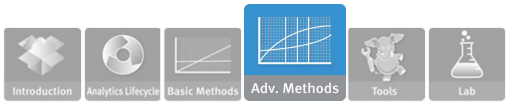
Adv. Methods

Module 4 – Advanced Analytics Theory and Methods

1



Module 4: Advanced Analytics – Theory and Methods

Upon completion of this module, you should be able to:

- Examine analytic needs and select an appropriate technique based on business objectives; initial hypotheses; and the data's structure and volume
- Apply some of the **more commonly used methods** in Analytics solutions
- **Explain the algorithms and the technical foundations** for the commonly used methods
- Explain the environment (use case) in which each technique can provide the most value
- Use appropriate diagnostic methods to validate the models created
- Use Python and in-database analytical functions to fit, score, and evaluate models

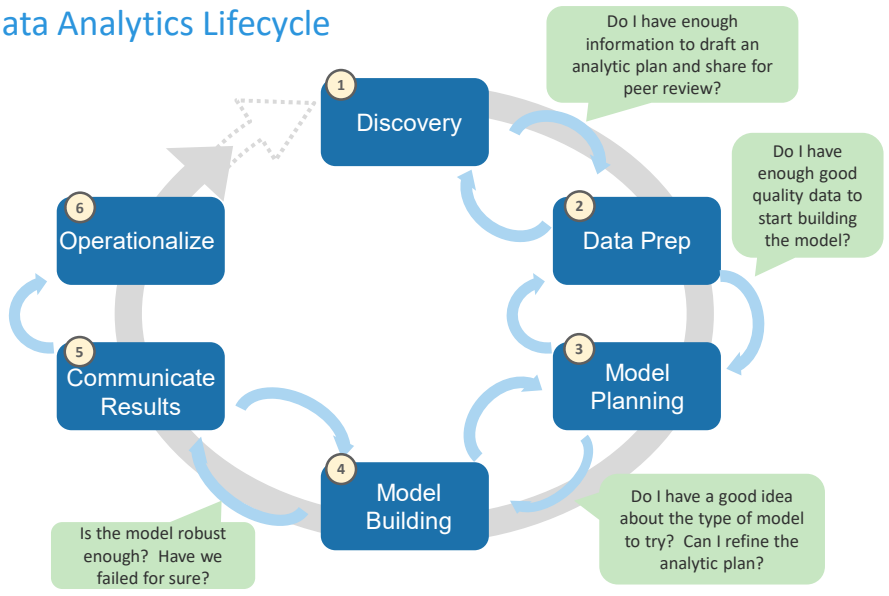
2

The Analytical Methods covered

- **Categorization (un-supervised)**
  1. K-Means clustering
  2. Association Rules
- **Regression**
  3. Linear
  4. Logistic
- **Classification (supervised)**
  5. Naive Bayesian classifier
  6. Decision Trees
  7. Time Series Analysis
  8. Text Analysis

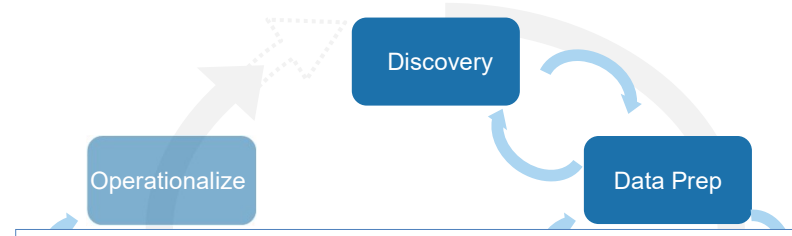
3

Data Analytics Lifecycle



4

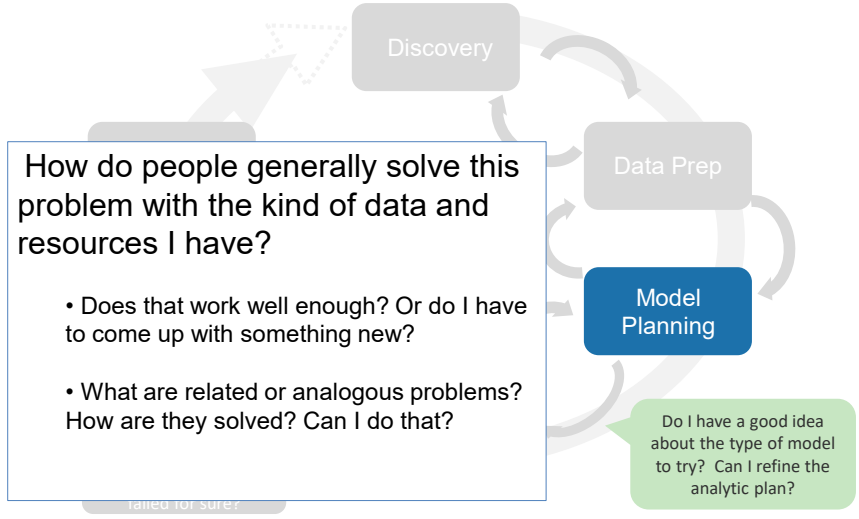
Applying the Data Analytics Lifecycle



- In a typical Data Analytics Problem - you would have gone through:
  - Phase 1 – Discovery - have the problem framed
  - Phase 2 – Data Preparation - have the data prepared
- Now you need to plan the model and determine the method to be used.

5

Phase 3 - Model Planning



How do people generally solve this problem with the kind of data and resources I have?

- Does that work well enough? Or do I have to come up with something new?
- What are related or analogous problems? How are they solved? Can I do that?

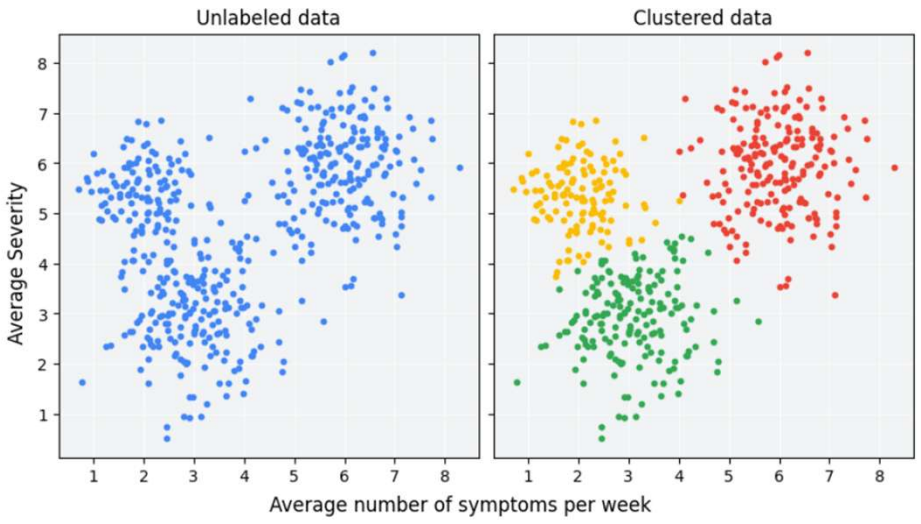
6

What Kind of Problem do I Need to Solve? How do I Solve it?

The Problem to Solve	The Category of Techniques	Covered in this Course
I want to group items by similarity. I want to find structure (commonalities) in the data	Clustering	**K-means clustering**
I want to discover relationships between actions or items	**Association Rules**	Apriori
I want to determine the relationship between the outcome and the input variables	Regression	**Linear Regression** **Logistic Regression**
I want to assign (known) labels to objects	Classification	**Naïve Bayes** **Decision Trees**
I want to find the structure in a temporal process I want to forecast the behavior of a temporal process	**Time Series Analysis**	ACF, PACF, ARIMA
I want to analyze my text data	**Text Analysis**	Regular expressions, Document representation (Bag of Words), TF-IDF

7

Clustering – K-means Clustering



8

Association Rules

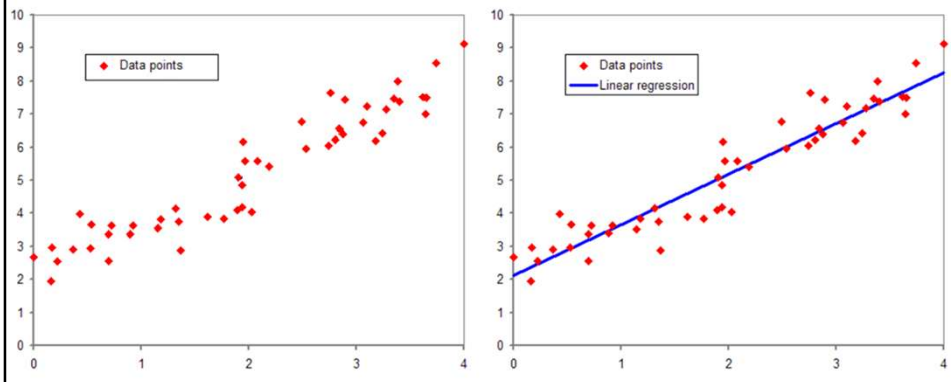
Market Basket Analysis is one of the key techniques used by large retailers to uncover associations between items.

Bread and Butter

Apriori Algorithm | Association Rule Mining | Finding Frequent Itemset |

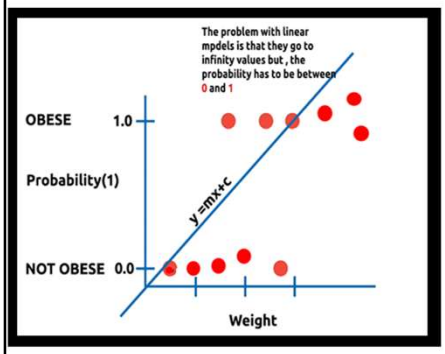
9

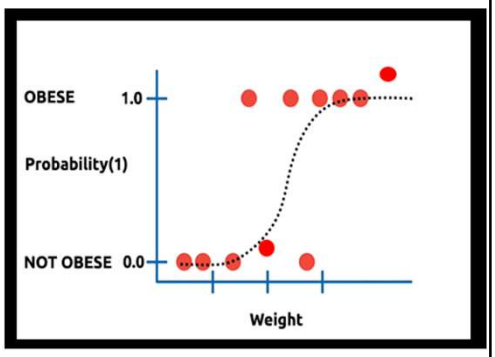
Linear Regression



10

Logistic Regression

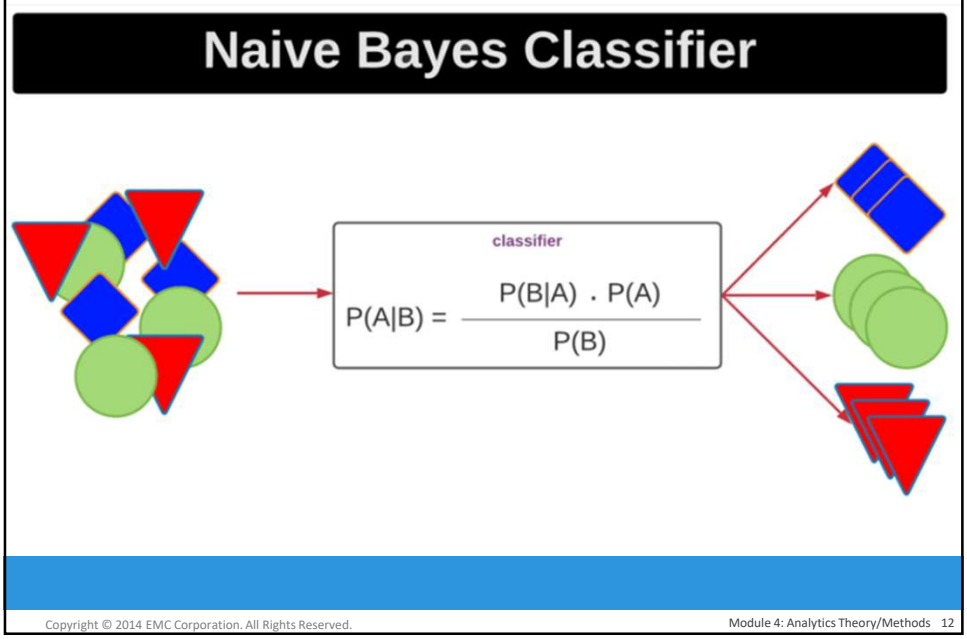




$$\text{Sigmoid} = \frac{1}{1 + e^{-(mx+c)}}$$

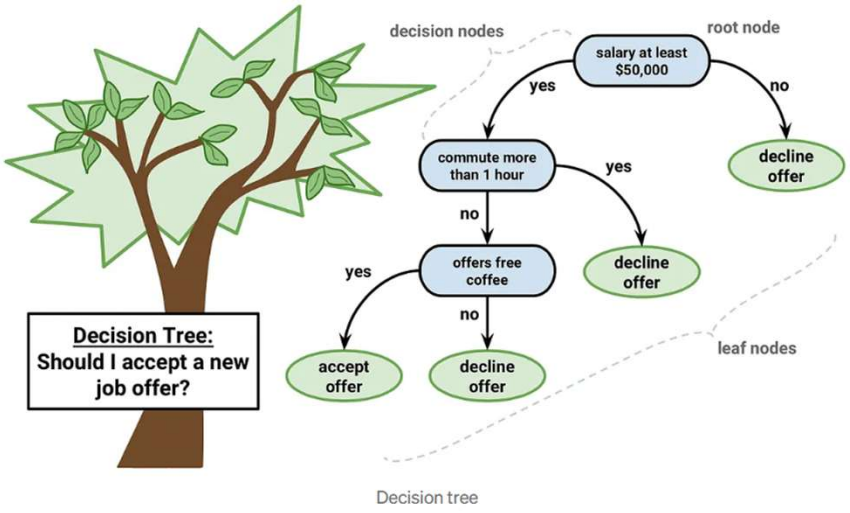
11

Classification - Naive Bayes



12

Classification – Decision Trees



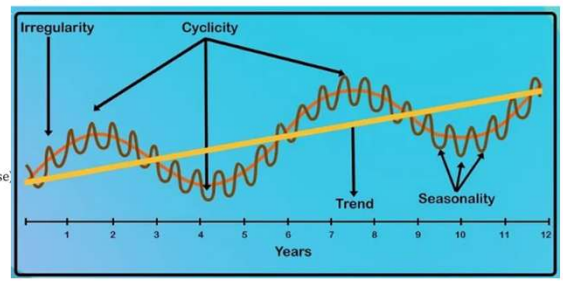
Decision tree

13

Time Series Analysis

1. **Descriptive:** Identify patterns in correlated data, such as trends and seasonal variations.
2. **Explanation:** These patterns may help in obtaining an understanding of the underlying forces and structure that produced the data.
3. **Forecasting:** In modelling the data, one may obtain accurate predictions of future (short-term) trends.
4. **Intervention analysis:** One can examine how (single) events have influenced the time series.
5. **Quality control:** Deviations on the time series may indicate problems in the process reflected by the data.

1. Trend
2. Seasonality
3. Cyclicals
4. Irregular/Random(Noise)

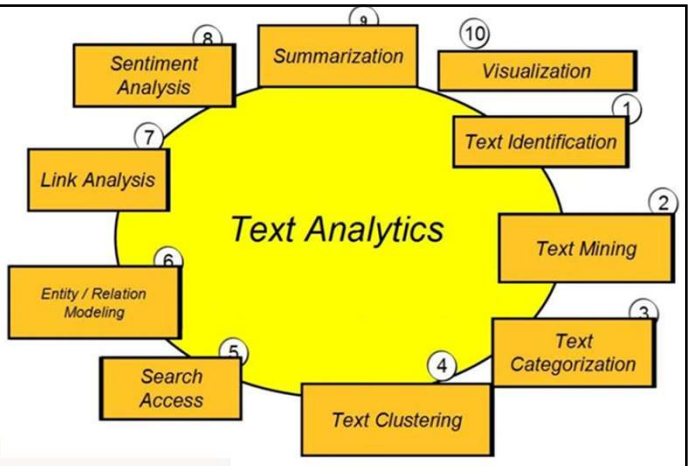


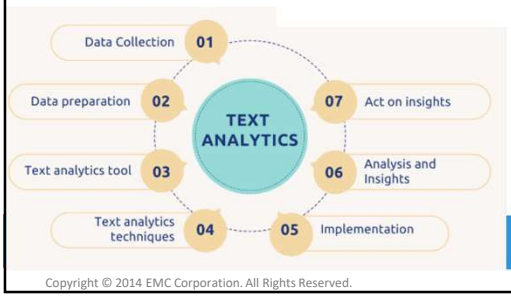
Economic Forecasting
Sales Forecasting
Budgetary Analysis
Stock Market Analysis
Yield Projections
Process and Quality Control
Inventory Studies
Workload Projections
Utility Studies
Census Analysis
Strategic Workforce Planning

14

Text Analysis

Text Analysis (Text Analytics or Text Mining) is the process of converting unstructured text data into meaningful data for analysis, to measure customer opinions, product reviews, feedback, to provide search facility, sentimental analysis and entity modeling to support fact based decision making.



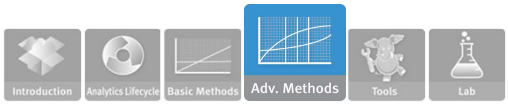


15

Why These Example Techniques?

- Most popular, frequently used:
  - Provide the foundation for Data Science skills on which to build
- Relatively easy for new Data Scientists to understand & comprehend
- Applicable to a broad range of problems in several verticals

16



Module 4: Advanced Analytics – Theory and Methods

Lesson 1: K-means Clustering

During this lesson the following topics are covered:

- Clustering – Unsupervised learning method
- K-means clustering:
  - Use cases
  - The algorithm
  - Determining the optimum value for K
  - Diagnostics to evaluate the effectiveness of the method
  - Reasons to Choose (+) and Cautions (-) of the method

17

Clustering

How do I group these documents by topic?

How do I group my customers by purchase patterns?

- Sort items into groups by similarity:
  - Items in a cluster are more similar to each other than they are to items in other clusters.
  - Need to detail the properties that characterize “similarity”
    - Or of distance, the "inverse" of similarity
- Not a predictive method; finds similarities, relationships
- Our Example: K-means Clustering

18

K-Means Clustering - What is it?

- Used for clustering numerical data, usually a set of measurements about objects of interest.
- Input: numerical. There must be a distance metric defined over the variable space.
  - Euclidian distance
- Output: The centers of each discovered cluster, and the assignment of each input datum to a cluster.
  - Centroid

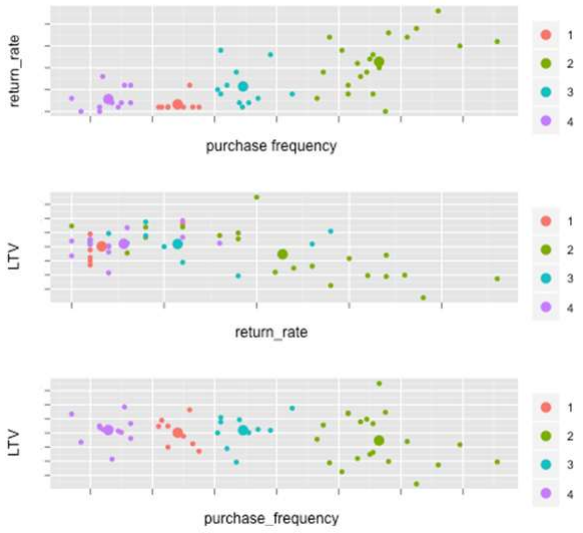
19

Use Cases

- Often an exploratory technique:
  - Discover structure in the data
  - Summarize the properties of each cluster
- Sometimes a prelude to classification:
  - "Discovering the classes“
- Examples
  - The height, weight and average lifespan of animals
  - Household income, yearly purchase amount in dollars, number of household members of customer households
  - Patient record with measures of BMI, HBA1C, HDL

20

Use-Case Example – On-line Retailer

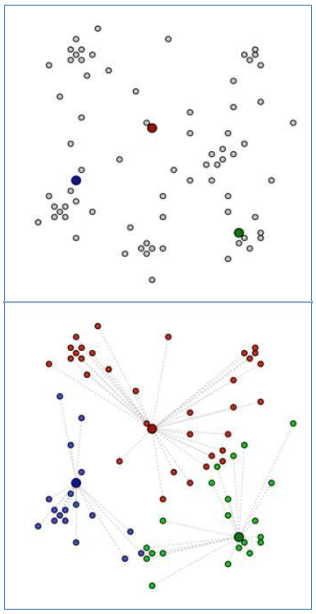


LTV – Lifetime Customer Value

21

The Algorithm

1. Choose *K*; then select K random "centroids"
   In our example, K=3
2. Assign records to the cluster with the closest centroid



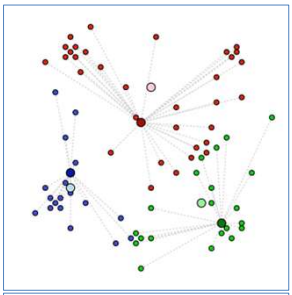
22

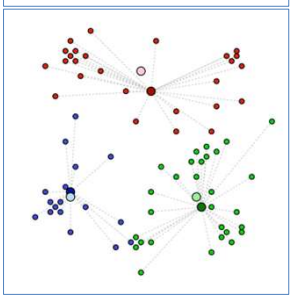
The Algorithm (Continued)

3. Recalculate the resulting centroids
   Centroid: the mean value of all the records in the cluster
4. Repeat steps 2 & 3 until record assignments no longer change

Model Output:

- The final cluster centers
- The final cluster assignments of the training data





23

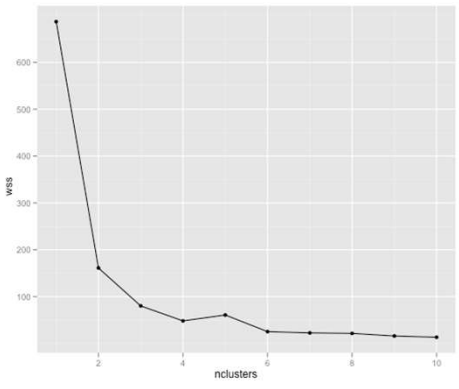
Picking K

Heuristic: find the "elbow" of the within-sum-of-squares (wss) plot as a function of K.

$$\text{WSS} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} |x_{ij} - c_i|^2$$

K: # of clusters
n_i: # points in i^{th} cluster
c_i: centroid of i^{th} cluster
x_{ij}: jth point of i^{th} cluster

"Elbows" at k=2,4,6



24

Diagnostics – Evaluating the Model

- Do the clusters look separated in at least some of the plots when you do pair-wise plots of the clusters?
  - Pair-wise plots can be used when there are not many variables
- Do you have any clusters with few data points?
  - Try decreasing the value of K
- Are there splits on variables that you would expect, but don't see?
  - Try increasing the value K
- Do any of the centroids seem too close to each other?
  - Try decreasing the value of K

25

K-Means Clustering - Reasons to Choose (+) and Cautions (-)

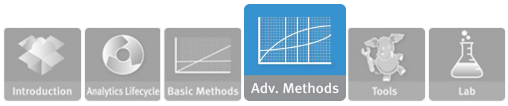
Reasons to Choose (+)	Cautions (-)
Easy to implement	Doesn't handle categorical variables
Easy to assign new data to existing clusters Which is the nearest cluster center?	Sensitive to initialization (first guess)
Concise output Coordinates the K cluster centers	Variables should all be measured on similar or compatible scales Not scale-invariant!
	K (the number of clusters) must be known or decided a priori Wrong guess: possibly poor results
	Tends to produce "round" equi-sized clusters. Not always desirable

26

Check Your Knowledge

1. Why do we consider K-means clustering as a unsupervised machine learning algorithm?
2. How do you use "pair-wise" plots to evaluate the effectiveness of the clustering?
3. Detail the four steps in the K-means clustering algorithm.
4. How do we use WSS to pick the value of K?
5. What is the most common measure of distance used with K-means clustering algorithms?
6. The attributes of a data set are "purchase decision (Yes/No), Gender (M/F), income group (<10K, 10-50K, >50K). Can you use K-means to cluster this data set?

27



Module 4: Advanced Analytics – Theory and Methods

Lesson 1: K-means Clustering - Summary

During this lesson the following topics were covered:

- Clustering – Unsupervised learning method
- What is K-means clustering
- Use cases with K-means clustering
- The K-means clustering algorithm
- Determining the optimum value for K
- Diagnostics to evaluate the effectiveness of K-means clustering
- Reasons to Choose (+) and Cautions (-) of K-means clustering

28

Lab Exercise 4: K-means Clustering

This Lab is designed to investigate and practice K-means Clustering.

After completing the tasks in this lab you should be able to:

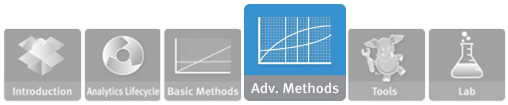
- Use R functions to create K-means Clustering models
- Use ODBC connection to the database and execute SQL statements and read database tables in an R environment
- Visualize the effectiveness of the K-means Clustering algorithm using graphic capabilities in R
- Use MADlib function for K-means Clustering

29

Lab Exercise 4: K-means Clustering - Workflow

1. Set the Working Directory
2. Establish the ODBC Connection
3. Open Connections to ODBC Database
4. Get Data from the Database
5. Read in the Data for Modeling
6. Execute the Model
7. Review the Output
8. Plot the Results
9. Find the Appropriate Number of Clusters
10. Close Connections to ODBC Database
11. Perform K-means Clustering Using In-database Analytics Method

30



Module 4: Advanced Analytics – Theory and Methods

Lesson 2: Association Rules

During this lesson the following topics are covered:

- Association Rules mining
- Apriori Algorithm
- Prominent use cases of Association Rules
- Support and Confidence parameters
- Lift and Leverage
- Diagnostics to evaluate the effectiveness of rules generated
- Reasons to Choose (+) and Cautions (-) of the Apriori algorithm

31

Association Rules

Which of my products tend to be purchased together?

What do other people like this person tend to like/buy/watch?

- Discover "interesting" relationships among variables in a large database
  - Rules of the form “If X is observed, then Y is also observed"
  - The definition of "interesting“ varies with the algorithm used for discovery
- Not a predictive method; finds similarities, relationships

32

Association Rules - Apriori

- Specifically designed for mining over transactions in databases
- Used over itemsets: sets of discrete variables that are linked:
  - Retail items that are purchased together
  - A set of tasks done in one day
  - A set of links clicked on by one user in a single session
- **Our Example: Apriori**

33

Apriori Algorithm - What is it?
Support

- Earliest of the association rule algorithms
- Frequent itemset: a set of items L that appears together "often enough":
  - Formally: meets a minimum support criterion
  - **Support**: the % of transactions that contain L
- Apriori Property: Any subset of a frequent itemset is also frequent
  - It has at least the support of its superset

34

Apriori Algorithm (Continued)
Confidence

- Iteratively grow the frequent itemsets from size 1 to size K (or until we run out of support).
  - Apriori property tells us how to prune the search space
- Frequent itemsets are used to find rules X->Y with a minimum **confidence**:
  - Confidence: The % of transactions that contain X, which also contain Y
- Output: The set of all rules X -> Y with minimum support and confidence

35

Lift and Leverage

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Support}(X \wedge Y)}{\text{Support}(X) * \text{Support}(Y)}$$

$$\text{Leverage}(X \rightarrow Y) = \text{Support}(X \wedge Y) - \text{Support}(X) * \text{Support}(Y)$$

36

Association Rules Implementations

- Market Basket Analysis
  - People who buy milk also buy cookies 60% of the time.
- Recommender Systems
  - "People who bought what you bought also purchased….".
- Discovering web usage patterns
  - People who land on page X click on link Y 76% of the time.

37

Use Case Example: Credit Records

Credit ID	Attributes
1	credit_good, female_married, job_skilled, home_owner, …
2	credit_bad, male_single, job_unskilled, renter, …

Minimum Support: 50%

Frequent Itemset	Support
credit_good	70%
male_single	55%
job_skilled	63%
home_owner	71%
home_owner, credit_good	53%

The itemset {home_owner, credit_good} has minimum support.

The possible rules are

credit_good -> home_owner

and

home_owner -> credit_good

38

Computing Confidence and Lift

Suppose we have 1000 credit records:

	free_housing	home_owner	renter	total
credit_bad	44	186	70	**300**
credit_good	64	527	109	**700**
	108	**713**	**179**	

713 home_owners, 527 have good credit.
home_owner -> credit_good has confidence 527/713 = 74%

700 with good credit, 527 of them are home_owners
credit_good -> home_owner has confidence 527/700 = 75%

The lift of these two rules is

0.527 / (0.700*0.713) = 1.055

39

A Sketch of the Algorithm

- If L_k is the set of frequent k-itemsets:
  - Generate the candidate set C_{k+1} by joining L_k to itself
  - Prune out the (k+1)-itemsets that don't have minimum support
    Now we have L_{k+1}
- We know this catches all the frequent (k+1)-itemsets by the apriori property
  - a (k+1)-itemset can't be frequent if any of its subsets aren't frequent
- Continue until we reach k_{max}, or run out of support
- From the union of all the L_k, find all the rules with minimum confidence

40

Step 1: 1-itemsets (L1)

- let min_support = 0.5
- 1000 credit records
- Scan the database
- Prune

Frequent Itemset	Count
credit_good	700
~~credit_bad~~	~~300~~
male_single	550
~~male_mar_or_wid~~	~~92~~
~~female~~	~~310~~
job_skilled	631
~~job_unskilled~~	~~200~~
home_owner	710
~~renter~~	~~179~~

41

Step 2: 2-itemsets (L2)

- Join L1 to itself
- Scan the database to get the counts
- Prune

Frequent Itemset	Count
~~credit_good, male_single~~	~~402~~
credit_good, job_skilled	544
credit_good, home_owner	527
~~male_single, job_skilled~~	~~340~~
~~male_single, home_owner~~	~~408~~
~~job_skilled, home_owner~~	~~452~~

42

Step 3: 3-itemsets

Frequent Itemset	Count
~~credit_good, job_skilled, home_owner~~	~~428~~

- We have run out of support.
- Candidate rules come from L2:
  - credit_good -> job_skilled
  - job_skilled -> credit_good
  - credit_good -> home_owner
  - home_owner -> credit_good

43

Finally: Find Confidence Rules

Rule	Set	Cnt	Set	Cnt	Confidence
IF credit_good THEN job_skilled	credit_good	700	credit_good AND job_skilled	544	544/700=77%
IF credit_good THEN home_owner	credit_good	700	credit_good AND home_owner	527	527/700=75%
IF job_skilled THEN credit_good	job_skilled	631	job_skilled AND credit_good	544	544/631=86%
IF home_owner THEN credit_good	home_owner	710	home_owner AND credit_good	527	527/710=74%

If we want confidence > 80%:
IF job_skilled THEN credit_good

44

Diagnostics

- Do the rules make sense?
  - What does the domain expert say?
- Make a "test set" from hold-out data:
  - Enter some market baskets with a few items missing (selected at random). Can the rules determine the missing items?
  - Remember, some of the test data may not cause a rule to fire.
- Evaluate the rules by lift or leverage.
  - Some associations may be coincidental (or obvious).

45

Apriori - Reasons to Choose (+) and Cautions (-)

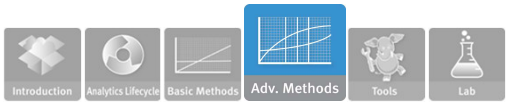
Reasons to Choose (+)	Cautions (-)
Easy to implement	Requires many database scans
Uses a clever observation to prune the search space •Apriori property	Exponential time complexity
Easy to parallelize	Can mistakenly find spurious (or coincidental) relationships •Addressed with Lift and Leverage measures

46

Check Your Knowledge

1. What is the Apriori property and how is it used in the Apriori algorithm?
2. List three popular use cases of the Association Rules mining algorithms.
3. What is the difference between Lift and Leverage. How is Lift used in evaluating the quality of rules discovered?
4. Define Support and Confidence
5. How do you use a "hold-out" dataset to evaluate the effectiveness of the rules generated?

47



Module 4: Advanced Analytics – Theory and Methods

Lesson 2: Association Rules - Summary

During this lesson the following topics were covered:

- Association Rules mining
- Apriori Algorithm
- Prominent use cases of Association Rules
- Support and Confidence parameters
- Lift and Leverage
- Diagnostics to evaluate the effectiveness of rules generated
- Reasons to Choose (+) and Cautions (-) of the Apriori algorithm

48

Lab Exercise 5 - Association Rules

- This Lab is designed to investigate and practice Association Rules.

After completing the tasks in this lab you should be able to:

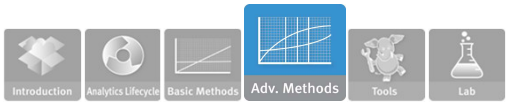
- Use R functions for Association Rule based models

49

Lab Exercise 5 - Association Rules - Workflow

1. Set the Working Directory and install the “arules” and "arulesViz" package
2. Read in the Data for Modeling
3. Review Transaction data
4. Plot Transactions
5. Mine the Association Rules
6. Read in Groceries dataset
7. Mine the Rules for the Groceries Data
8. Extract Rules with Confidence > 0.8

50



Module 4: Advanced Analytics – Theory and Methods

Lesson 3: Linear Regression

During this lesson the following topics are covered:

- General description of regression models
- Technical description of a linear regression model
- Common use cases for the linear regression model
- Interpretation and scoring with the linear regression model
- Diagnostics for validating the linear regression model
- The Reasons to Choose (+) and Cautions (-) of the linear regression model

51

Regression

- Regression focuses on the relationship between an outcome and its input variables.
  - Provides an estimate of the outcome based on the input values.
  - Models how changes in the input variables affect the outcome.
- The outcome can be continuous or discrete.
- Possible use cases:
  - Estimate the lifetime value (LTV) of a customer and understand what influences LTV.
  - Estimate the probability that a loan will default and understand what leads to default.
- **Our approaches: linear regression and logistic regression**

52

Linear Regression

- Used to estimate a continuous value as a linear (additive) function of other variables
  - Income as a function of years of education, age, and gender
  - House sales price as function of square footage, number of bedrooms/bathrooms, and lot size
- Outcome variable is continuous.
- Input variables can be continuous or discrete.
- Model Output:
  - A set of estimated coefficients that indicate the relative impact of each input variable on the outcome
  - A linear expression for estimating the outcome as a function of input variables

53

Linear Regression Model

~~$y = \beta_0 + \beta_0 x_1 + \beta_0 x_1 + \cdots + \beta_{p-1} x_{p-1} + \varepsilon$~~

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{p-1} x_{p-1} + \varepsilon$$

where y is the outcome variable

x_j are the input variables, for $j = 1,2,\ldots,p-1$

β_0 is the value of y when each x_j equals zero

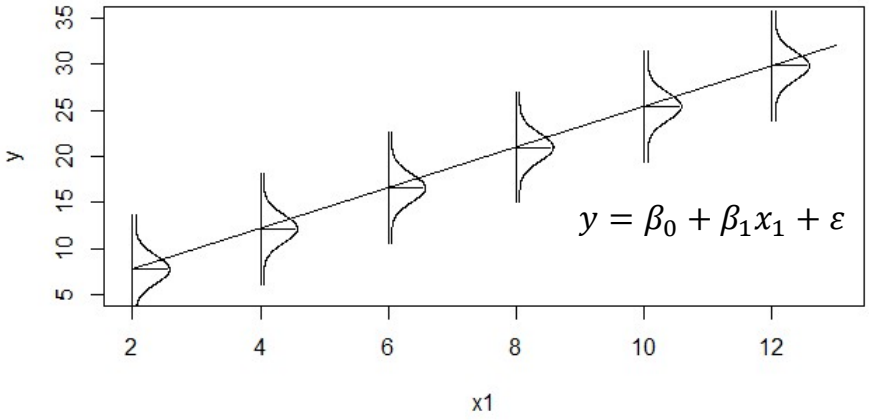
β_j is the change in y based on a unit change in x_j

$\varepsilon \sim N(0, \sigma^2)$ and the ε's are independent of each other

54

Example: Linear Regression with One Input Variable

- x_1 - the number of employees reporting to a manager
- y - the hours per week spent in meetings by the manager



55

Representing Categorical Attributes

$$y = \beta_0 + \beta_1 employees + \beta_2 finance + \beta_3 mfg + \beta_4 sales + \varepsilon$$

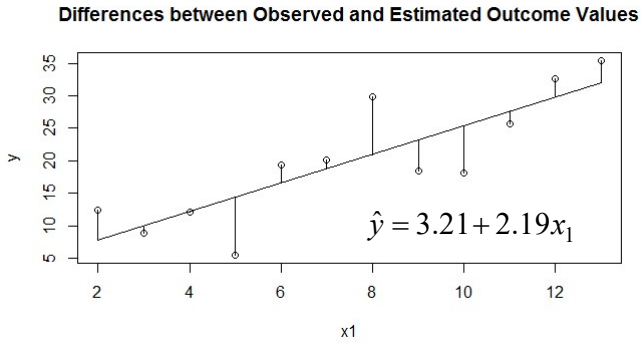
Possible Situation	Input Variables
Finance manager with 8 employees	(8,1,0,0)
Manufacturing manager with 8 employees	(8,0,1,0)
Sales manager with 8 employees	(8,0,0,1)
Engineering manager with 8 employees	(8,0,0,0)

- For a categorical attribute with *m* possible values
  - Add *m-1* binary (0/1) variables to the regression model
  - The remaining category is represented by setting the *m-1* binary variables equal to zero

56

Fitting a Line with Ordinary Least Squares (OLS)

- Choose the line that minimizes: $\sum_{i=1}^{n}[y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_{p-1} x_{i,p-1})]^2$
- Provides the coefficient estimates, denoted b_j



57

Interpreting the Estimated Coefficients, b_j

$$\hat{y} = 4.0 + 2.2employees + 0.5\,finance - 1.9mfg + 0.6sales$$

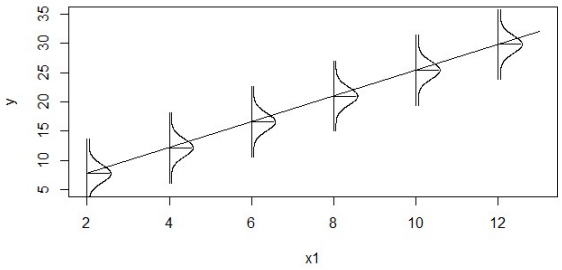
- Coefficients for numeric input variables
  - Change in outcome due to a unit change in input variable*
  - Example: $b_1 = 2.2$
    - Extra 2.2 hrs/wk in meetings for each additional employee managed*
- Coefficients for binary input variables
  - Represent the additive difference from the reference level *
  - Example: $b_2 = 0.5$
    - Finance managers meet 0.5 hr/wk more than engineering managers do*
- Statistical significance of each coefficient
  - Are the coefficients significantly different from zero?
  - For small *p*-values (say < 0.05), the coefficient is statistically significant

* when all other input values remain the same

58

Diagnostics – Examining Residuals

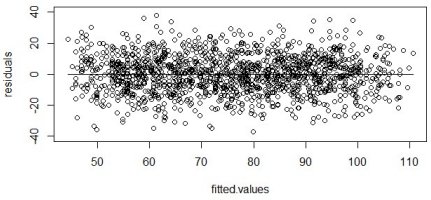
- Residuals
  - Differences between the observed and estimated outcomes
  - The observed values of the error term, ε, in the regression model
  - Expressed as: $e_i = y_i - \hat{y}_i \qquad for\ i = 1,2...,n$
- Errors are assumed to be normally distributed with
  - A mean of zero
  - Constant variance



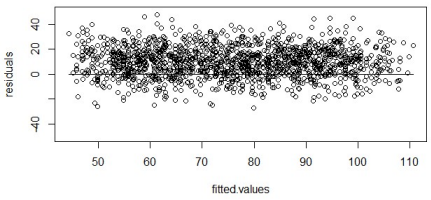
59

Diagnostics – Plotting Residuals

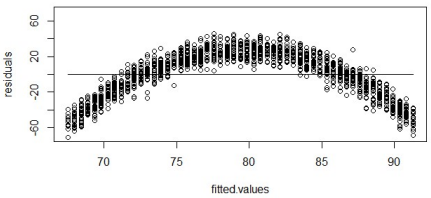
Ideal Residual Plot



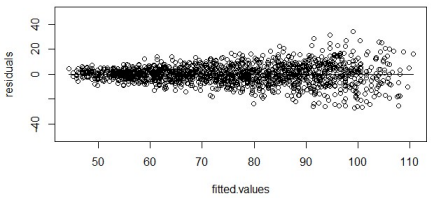
Non-centered



Quadratic Trend



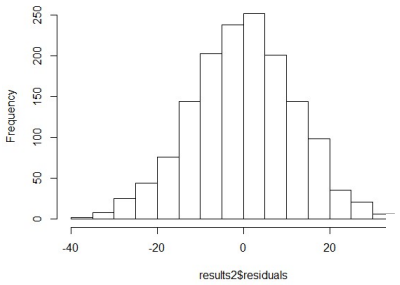
Non-constant Variance



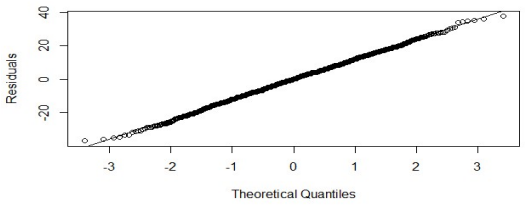
60

Diagnostics – Residual Normality Assumption

Ideal Histogram



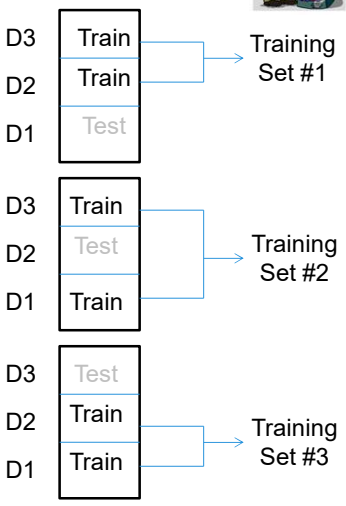
Ideal Q-Q Plot



61

Diagnostics – Using Hold-out Data

- Hold-out data
  - Training and testing datasets
  - Does the model predict well on data it hasn't seen?
- N-fold cross validation
  - Partition the data into N groups.
  - Holding out each group,
    - Fit the model
    - Calculate the residuals on the group
  - Estimated prediction error is the average over all the residuals.



62

Diagnostics – Other Considerations

- R^2
  - The fraction of the variability in the outcome variable explained by the fitted regression model.
  - Attains values from 0 (poorest fit) to 1 (perfect fit)
- Identify correlated input variables
  - Pair-wise scatterplots
  - Sanity check the coefficients
    - Are the magnitudes excessively large?
    - Do the signs make sense?
- $R^2 = 1 - SS_{err}/SS_{tot}$
- ~ 1

$$SS_{err} = \sum (y - y_{pred})^2$$

$$SS_{tot} = \sum (y - y_{mean})^2$$

63

Linear Regression - Reasons to Choose (+) and Cautions (-)

Reasons to Choose (+)	Cautions (-)
Concise representation (the coefficients)	Does not handle missing values well
Robust to redundant or correlated variables Lose some explanatory value	Assumes that each variable affects the outcome linearly and additively Variable transformations and modeling variable interactions can alleviate this A good idea to take the log of monetary amounts or any variable with a wide dynamic range
Explanatory value Relative impact of each variable on the outcome	Does not easily handle variables that affect the outcome in a discontinuous way Step functions
Easy to score data	Does not work well with categorical attributes with a lot of distinct values For example, ZIP code

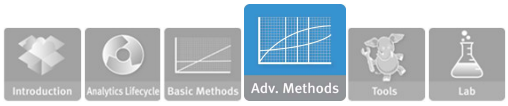
64

Check Your Knowledge

Your Thoughts?

1. How is the measure of significance used in determining the explanatory value of a driver (input variable) with linear regression models?
2. Detail the challenges with categorical values in linear regression model.
3. Describe N-Fold cross validation method used for diagnosing a fitted model.
4. List two use cases of linear regression models.
5. List and discuss two standard checks that you will perform on the coefficients derived from a linear regression model.

65



Module 4: Advanced Analytics – Theory and Methods

Lesson 3: Linear Regression - Summary

During this lesson the following topics were covered:

- General description of regression models
- Technical description of a linear regression model
- Common use cases for the linear regression model
- Interpretation and scoring with the linear regression model
- Diagnostics for validating the linear regression model
- The Reasons to Choose (+) and Cautions (-) of the linear regression model

66

Lab Exercise 6: Linear Regression

This Lab is designed to investigate and practice Linear Regression.

After completing the tasks in this lab you should be able to:

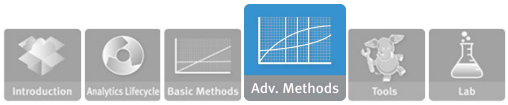
- Use R functions for Linear Regression (Ordinary Least Squares – OLS)
- Predict the dependent variables based on the model
- Investigate different statistical parameter tests that measure the effectiveness of the model

67

Lab Exercise 6: Linear Regression - Workflow

1. Set Working directory
2. Generate random data to model
3. Generate the OLS model
4. Print and visualize the results
5. Generate summary outputs
6. Introduce a slight non-linearity
7. Perform in-database analysis of linear regression

68



Module 4: Advanced Analytics – Theory and Methods

Lesson 4: Logistic Regression

During this lesson the following topics are covered:

- Technical description of a logistic regression model
- Common use cases for the logistic regression model
- Interpretation and scoring with the logistic regression model
- Diagnostics for validating the logistic regression model
- Reasons to Choose (+) and Cautions (-) of the logistic regression model

69

Logistic Regression

- Used to estimate the probability that an event will occur as a function of other variables
  - The probability that a borrower will default as a function of his credit score, income, the size of the loan, and his existing debts
- Can be considered a classifier, as well
  - Assign the class label with the highest probability
- Input variables can be continuous or discrete
- Output:
  - A set of coefficients that indicate the relative impact of each driver
  - A linear expression for predicting the log-odds ratio of outcome as a function of drivers. (Binary classification case)
    - Log-odds ratio easily converted to the probability of the outcome

70

Logistic Regression Use Cases

- The preferred method for many binary classification problems:
  - Especially if you are interested in the probability of an event, not just predicting the "yes or no“
  - Try this first; if it fails, then try something more complicated
- Binary Classification examples:
  - The probability that a borrower will default
  - The probability that a customer will churn
- Multi-class example
  - The probability that a politician will vote yes/vote no/not show up to vote on a given bill

71

Logistic Regression Model - Example

$$\text{default} = f(\text{creditScore}, \text{income}, \text{loanAmt}, \text{existingDebt})$$

- Training data: default is 0/1
  - default=1 if loan defaulted
- The model will return the probability that a loan with given characteristics will default
- If you only want a "yes/no" answer, you need a threshold
  - The standard threshold is 0.5

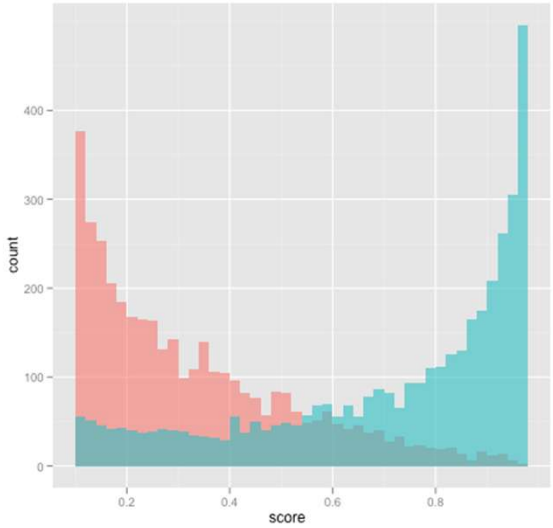
72

Logistic Regression- Visualizing the Model

Overall fraction of default: ~20%

Logistic regression returns a score that estimates the probability that a borrower will default

The graph compares the distribution of defaulters and non-defaulters as a function of the model's predicted probability, for borrowers scoring higher than 0.1

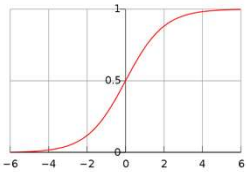


Blue=defaulters

73

Technical Description (Binary Case)

$$ln\left(\frac{P(y=1)}{1-P(y=1)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \ldots + \beta_p x_{p-1}$$



- y=1 is the case of interest: 'TRUE'
- LHS is called logit(P(y=1))
  - hence, "logistic regression"
- logit(P(y=1)) is inverted by the sigmoid function
  - standard packages can return probability for you
- Categorical variables are expanded as with linear regression
- Iterative solution to obtain coefficient estimates, denoted b_j
  - "Iteratively re-weighted least squares"

74

Interpreting the Estimated Coefficients, b_i

- Invert the logit expression:

$$\frac{P(y=1)}{1-P(y=1)} = \exp(\sum_{j=0}^{K} b_j x_j)$$

$$= \prod_{j=0}^{K} \exp(b_j x_j)$$

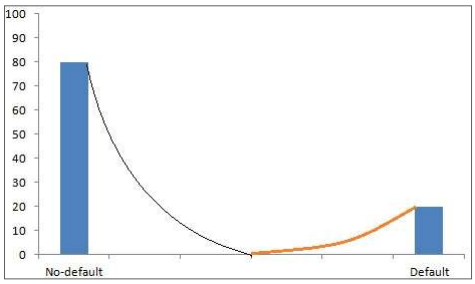
- exp(b_j) tells us how the odds-ratio of y=1 changes for every unit change in x_j
- Example: $b_{creditScore}$ = -0.69
  - exp($b_{creditScore}$) = 0.5 = 1/2
  - for the same income, loan, and existing debt, the odds-ratio of default is halved for every point increase in credit score
- Standard packages return the significance of the coefficients in the same way as in linear regression

75

An Interesting Fact About Logistic Regression

"The probability mass equals the counts"

- If 20% of our loan risk training set defaults
  - The sum of all the training set scores will be 20% of the number of training examples
- If 40% of applicants with income < $50,000 default
  - The sum of all the training set scores of people in this income category will be 40% of the number of examples in this income category



76

Diagnostics

- Hold-out data:
  - Does the model predict well on data it hasn't seen?
- N-fold cross-validation: Formal estimate of generalization error
- "Pseudo-R^2" : 1 – (deviance/null deviance)
  - Deviance, null deviance both reported by most standard packages
  - The fraction of "variance" that is explained by the model
  - Used the way R^2 is used

77

Diagnostics (Cont.)

- Sanity check the coefficients
  - Do the signs make sense? Are the coefficients excessively large?
    - Wrong sign is an indication of correlated inputs, but doesn't necessarily affect predictive power.
    - Excessively large coefficient magnitudes may indicate strongly correlated inputs; you may want to consider eliminating some variables, or using regularized regression techniques.
    - Infinite magnitude coefficients could indicate a variable that strongly predicts a subset of the output (and doesn't predict well on the rest).
      - Try a Decision Tree on that variable, to see if you should segment the data before regressing.

78

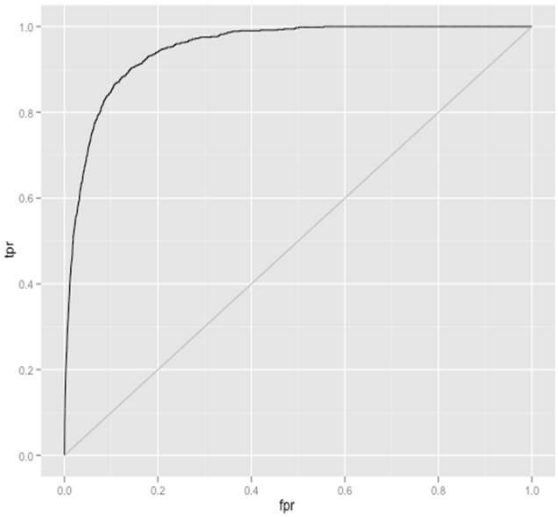
Diagnostics: ROC Curve

$$\text{FPR} = \frac{\#\text{ false positives}}{\text{all negatives}}$$

$$\text{TPR} = \frac{\#\text{ true positives}}{\text{all positives}}$$

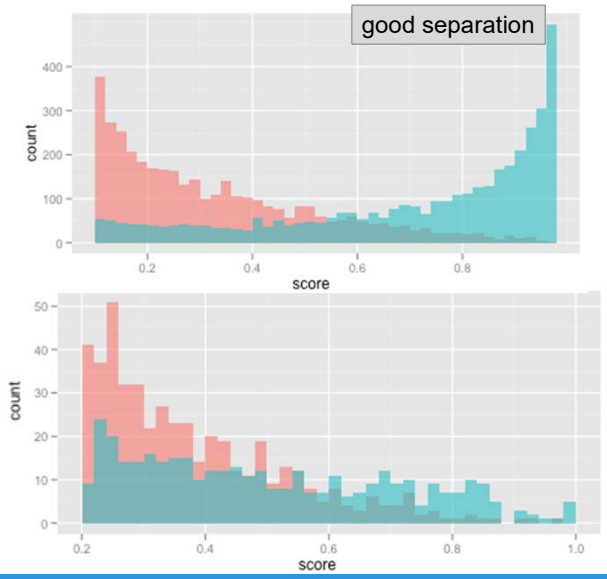
Area under the curve (AUC) tells you how well the model predicts. (Ideal AUC = 1)

For logistic regression, ROC (Receiver Operating Characteristics) curve can help set classifier threshold



79

Diagnostics: Plot the Histograms of Scores



80

Logistic Regression - Reasons to Choose (+) and Cautions (-)

Reasons to Choose (+)	Cautions (-)
Explanatory value: Relative impact of each variable on the outcome in a more complicated way than linear regression	Does not handle missing values well
Robust with redundant variables, correlated variables Lose some explanatory value	Assumes that each variable affects the log-odds of the outcome linearly and additively Variable transformations and modeling variable interactions can alleviate this A good idea to take the log of monetary amounts or any variable with a wide dynamic range
Concise representation with the the coefficients	Cannot handle variables that affect the outcome in a discontinuous way. Step functions
Easy to score data	Doesn't work well with discrete drivers that have a lot of distinct values For example, ZIP code
Returns good probability estimates of an event	
Preserves the summary statistics of the training data "The probabilities equal the counts"	

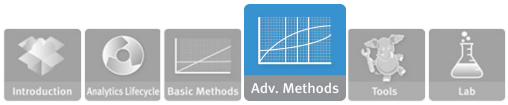
81

Check Your Knowledge

Your Thoughts?

1. What is a logit and how do we compute class probabilities from the logit?
2. How is ROC curve used to diagnose the effectiveness of the logistic regression model?
3. What is Pseudo R^2 and what does it measure in a logistic regression model?
4. How do you describe a binary class problem?
5. Compare and contrast linear and logistic regression methods.

82



Module 4: Advanced Analytics – Theory and Methods

Lesson 4: Logistic Regression - Summary

During this lesson the following topics were covered:

- Technical description of a logistic regression model
- Common use cases for the logistic regression model
- Interpretation and scoring with the logistic regression model
- Diagnostics for validating the logistic regression model
- Reasons to Choose (+) and Cautions (-) of the logistic regression model

83

Lab Exercise 7: Logistic Regression

This Lab is designed to investigate and practice Logistic Regression.

After completing the tasks in this lab you should be able to:

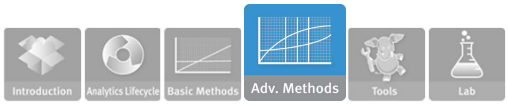
- Use R functions for Logistic Regression – (*also known as Logit*)
- Predict the dependent variables based on the model
- Investigate different statistical parameter tests that measure the effectiveness of the model

84

Lab Exercise 7: Logistic Regression - Workflow

1. Define the problem and review input data
2. Set the Working Directory
3. Read in and examine the data
4. Build and review logistic regression model
5. Review the results and interpret the coefficients
6. Visualize the model using the Plot function
7. Use Relevel function to re-level the Price factor with value 30 as the base reference
8. Plot the ROC curve
9. Predict Outcome given Age and Income
10. Predict outcome for a sequence of Age values at price 30 and mean income
11. Predict outcome for a sequence of income at price 30 and mean age
12. Use logistic regression as a classifier

85



Module 4: Advanced Analytics – Theory and Methods

Lesson 5: Naïve Bayesian Classifiers

During this lesson the following topics are covered:

- Naïve Bayesian Classifier
- Theoretical foundations of the classifier
- Use cases
- Evaluating the effectiveness of the classifier
- The Reasons to Choose (+) and Cautions (-) with the use of the classifier

86

Classifiers

Where in the catalog should I place this product listing?
Is this email spam?
Is this politician Democrat/Republican/Green?

- Classification: assign labels to objects.
- Usually supervised: training set of pre-classified examples.
- Our examples:
  - Naïve Bayesian
  - Decision Trees
  - (and Logistic Regression)

87

Naïve Bayesian Classifier

- Determine the most probable class label for each object
  - Based on the observed object attributes
    - Naïvely assumed to be conditionally independent of each other
  - Example:
    - Based on the objects attributes {shape, color, weight}
    - A given object that is {spherical, yellow, < 60 grams}, may be classified (labeled) as a tennis ball
  - Class label probabilities are determined using Bayes' Law
- Input variables are discrete
- Output:
  - Probability score – proportional to the true probability
  - Class label – based on the highest probability score

88

Naïve Bayesian Classifier - Use Cases

- Preferred method for many text classification problems.
  - Try this first; if it doesn't work, try something more complicated
- Use cases
  - Spam filtering, other text classification tasks
  - Fraud detection

- https://en.wikipedia.org/wiki/Naive_Bayes_spam_filtering

- https://eprints.soton.ac.uk/268483/
- https://convo.co.uk/

89

Building a Training Dataset to Predict Good or Bad Credit

- Predict the credit behavior of a credit card applicant from applicant's attributes:
  - Personal status
  - Job type
  - Housing type
  - Savings amount
- These are all categorical variables and are better suited to Naïve Bayesian Classifier than to logistic regression.

personal_status	job	housing	savings_status	credit_class
male single	skilled	own	no known savings	good
female div/dep/mar	skilled	own	<100	bad
male single	unskilled resident	own	<100	good
male single	skilled	for free	<100	good
male single	skilled	for free	<100	bad
male single	unskilled resident	for free	no known savings	good
male single	skilled	own	500<=X<1000	good
male single	high qualif/self emp/mgm	rent	<100	good
male div/sep	unskilled resident	own	>=1000	good
male mar/wid	high qualif/self emp/mgm	own	<100	bad
female div/dep/mar	skilled	rent	<100	bad
female div/dep/mar	skilled	rent	<100	bad
female div/dep/mar	skilled	own	<100	good
male single	unskilled resident	own	<100	bad
female div/dep/mar	skilled	rent	<100	good
female div/dep/mar	unskilled resident	own	100<=X<500	bad
male single	skilled	own	no known savings	good
male single	skilled	own	no known savings	good
female div/dep/mar	high qualif/self emp/mgm	for free	<100	bad
male single	skilled	own	500<=X<1000	good
male single	skilled	own	<100	good
male single	skilled	rent	500<=X<1000	good
male single	unskilled resident	rent	<100	good
male single	skilled	own	100<=X<500	good
male mar/wid	skilled	own	no known savings	good
male single	unskilled resident	own	<100	good
male mar/wid	unskilled resident	own	<100	good

90

Technical Description - Bayes' Law

$$P(C \mid A) = \frac{P(A \cap C)}{P(A)} = \frac{P(A \mid C)P(C)}{P(A)}$$

Reverend Thomas Bayes

- C is the class label:
  - $C \in \{C_1, C_2, \ldots C_n\}$
- A is the observed object attributes
  - $A = (a_1, a_2, \ldots a_m)$
- P(C | A) is the probability of C given A is observed
  - Called the conditional probability

91

An example using Bayes Law:

He has determined that, if he checks in for his flight at least two hours early, the probability that he will get the upgrade is .75; otherwise, the probability that he will get the upgrade is .35. With his busy schedule, he checks in at least two hours before his flight only 40% of the time. Suppose John didn't receive an upgrade on his most recent attempt. What is the probability that he arrived late?

C = John arrives late
A = John did not receive an upgrade
P(C) = Probability John arrives late = .6
P(A) = Probability John did not receive an upgrade = 1 – (.4 x .75 + .6 x .35) = 1 -.51 = .49
P(A|C) = Probability that John did not receive an upgrade given that he arrived late = 1 - .35 = .65
P(C|A) = Probability that John arrived late given that he did not receive his upgrade
= P(A|C)P(C)/P(A) = (.65 x .6)/.49 = .7959 ~ 0.80 (approx)

92

A second example of Bayes Law:

Let us assume that a disease exists and a test for it has been developed.

We know the following probabilities:

P(C) = probability of having the disease = .05
P(¬C) = probability of not having the disease = .95
P(A|C) = probability of testing positive, if having the disease = .95
P(A|¬C) = probability of testing positive, if not having the disease = .1

In order to find if this a reliable test, we need to solve for the probability of having the disease given you have a positive test result, P(C|A). From Bayes' Law, P(C|A) = P(A|C) P(C) / P(A). We need to compute P(A).

P(A) = probability of testing positive = P(C) * P(A|C) + P(¬C) * P(A|¬C) = .05 * .95 + .95 * .1 = 0.1425

- So, P(C|A) = P(A|C) P(C) / P(A) = (.95 * .05) / 0.1425 = 1/3, which means that the probability of a patient having the disease given that the patient tested positive is only one third. This may not be a good test!

93

A second example of Bayes Law:

Illustrating with hard numbers, lets fill the following test / disease matrix for a population of 10,000 patients.

	Have Disease	Do not have Disease	Totals
Test positive	10000 * (.05*.95)	10000 * (.95*.1)	
	475	950	1425
Test negative	10000 * (.05*.05)	10000 * (.95*.9)	
	25	8550	8575

475 / 1425 = 1 / 3

94

Technical Description - Bayes' Law

$$P(C \mid A) = \frac{P(A \cap C)}{P(A)} = \frac{P(A \mid C)P(C)}{P(A)}$$

Reverend Thomas Bayes

- C is the class label:
  - $C \in \{C_1, C_2, \ldots C_n\}$
- A is the observed object attributes
  - $A = (a_1, a_2, \ldots a_m)$
- P(C | A) is the probability of C given A is observed
  - Called the conditional probability

95

Apply the Naïve Assumption and Remove a Constant

- For observed attributes $A = (a_1, a_2, \ldots a_m)$, we want to compute

$$P(C_i \mid A) = \frac{P(a_1, a_2, \ldots, a_m \mid C_i)P(C_i)}{P(a_1, a_2, \ldots, a_m)} \qquad i = 1, 2, \ldots, n$$

  and assign the classifier, C_i, with the largest $P(C_i \mid A)$

- Two simplifications to the calculations
  - Apply naïve assumption - each a_j is conditionally independent of each other, then

$$P(a_1, a_2, \ldots, a_m \mid C_i) = P(a_1 \mid C_i)P(a_2 \mid C_i) \cdots P(a_m \mid C_i) = \prod_{j=1}^{m} P(a_j \mid C_i)$$

  - Denominator $P(a_1, a_2, \ldots a_m)$ is a constant and can be ignored

96

Building a Naïve Bayesian Classifier

- Applying the two simplifications

$$P(C_i \mid a_1, a_2, ..., a_m) \propto \left(\prod_{j=1}^{m} P(a_j \mid C_i) \right) P(C_i) \qquad i = 1, 2, ..., n$$

- To build a Naïve Bayesian Classifier, collect the following statistics from the training data:
  - $P(C_i)$ for all the class labels.
  - $P(a_j \mid C_i)$ for all possible a_j and C_i
  - Assign the classifier label, C_i, that maximizes the value of

$$\left(\prod_{j=1}^{m} P(a_j \mid C_i) \right) P(C_i) \qquad i = 1, 2, ..., n$$

97

Naïve Bayesian Classifiers for the Credit Example

- Class labels: {good, bad}
  - P(good) = 0.7
  - P(bad) = 0.3
- Conditional Probabilities
  - P(own|bad) = 0.62
  - P(own|good) = 0.75
  - P(rent|bad) = 0.23
  - P(rent|good) = 0.14
  - … and so on

P_{bad}= 8 / 27 = 0.30

P_{good}= 19 / 27 = 0.70

P(own|good)=P(own (int) good)/P(good)
= (13 / 27) / (19 / 27)) = 13/19 = here 68%

personal_status	job	housing	savings_status	credit_class
male single	skilled	own	no known savings	good
female div/dep/mar	skilled	own	<100	bad
male single	unskilled resident	own	<100	good
male single	skilled	for free	<100	good
male single	skilled	for free	<100	bad
male single	unskilled resident	for free	no known savings	good
male single	skilled	own	500<=X<1000	good
male single	high qualif/self emp/mgm	rent	<100	good
male div/sep	unskilled resident	own	>=1000	good
male mar/wid	high qualif/self emp/mgm	own	<100	bad
female div/dep/mar	skilled	rent	<100	bad
female div/dep/mar	skilled	rent	<100	bad
female div/dep/mar	skilled	own	<100	good
male single	unskilled resident	own	<100	bad
female div/dep/mar	skilled	rent	<100	good
female div/dep/mar	unskilled resident	own	100<=X<500	bad
male single	skilled	own	no known savings	good
male single	skilled	own	no known savings	good
female div/dep/mar	high qualif/self emp/mgm	for free	<100	bad
male single	skilled	own	500<=X<1000	good
male single	skilled	own	<100	good
male single	skilled	rent	500<=X<1000	good
male single	unskilled resident	rent	<100	good
male single	skilled	own	100<=X<500	good
male mar/wid	skilled	own	no known savings	good
male single	unskilled resident	own	<100	good
male single	unskilled resident	own	<100	good

There are 1000 samples available, only 28 samples are shown here…

98

Naïve Bayesian Classifier for a Particular Applicant

- Given applicant attributes of
  A= {female single,
  owns home,
  self-employed,
  savings > $1000}
- Since P(good|A) > (bad|A), assign the applicant the label "good" credit

a_j	C_i	$P(a_j \mid C_i)$
female single	good	0.28
female single	bad	0.36
own	good	0.75
own	bad	0.62
self emp	good	0.14
self emp	bad	0.17
savings>1K	good	0.06
savings>1K	bad	0.02

P(good|A) ~ (0.28*0.75*0.14*0.06)*0.7 = 0.0012

P(bad|A) ~ (0.36*0.62*0.17*0.02)*0.3 = 0.0002

99

Naïve Bayesian Implementation Considerations

- Numerical underflow
  - Resulting from multiplying several probabilities near zero
  - Preventable by computing the logarithm of the products
- Zero probabilities due to unobserved attribute/classifier pairs
  - Resulting from rare events
  - Handled by smoothing (adjusting each probability by a small amount)
- Assign the classifier label, C_i, that maximizes the value of

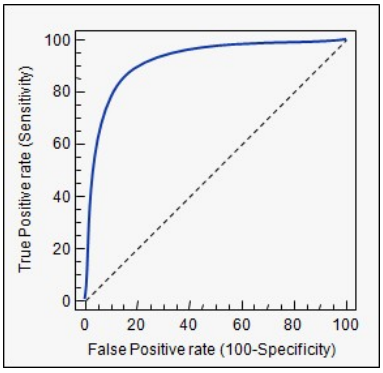
$$\left(\sum_{j=1}^{m} \log P'(a_j \mid C_i)\right) + \log P(C_i)$$

where i = 1,2,…,n and
P′ denotes the adjusted probabilities

100

Diagnostics

- Hold-out data
  - How well does the model classify new instances?
- Cross-validation
- ROC curve/AUC



101

Diagnostics: Confusion Matrix

		Prediction	
Actual Class	good	bad	
good	671	29	700
bad	38	262	300
	709	291	1000

true positives (TP)

false negatives (FN)

false positives (FP)

true negatives (TN)

Overall success rate (or accuracy):
(TP + TN) / (TP+TN+FP+FN) = (671+262)/1000 ≈ 0.93

TPR: TP / (TP + FN) = 671 / (671+29) = 671/700 ≈ 0.96
FPR: FP / (FP + TN) = 38 / (38 + 262) = 38/300 ≈ 0.13
FNR: FN / (TP + FN) = 29 / (671 + 29) = 29/700 ≈ 0.04

Precision: TP/ (TP + FP) = 671/709 ≈ 0.95
Recall (or TPR): TP / (TP + FN) ≈ 0.96

102

Naïve Bayesian Classifier - Reasons to Choose (+) and Cautions (-)

Reasons to Choose (+)	Cautions (-)
Handles missing values quite well	Numeric variables have to be discrete (categorized) Intervals
Robust to irrelevant variables	Sensitive to correlated variables "Double-counting"
Easy to implement	Not good for estimating probabilities Stick to class label or yes/no
Easy to score data	
Resistant to over-fitting	
Computationally efficient Handles very high dimensional problems Handles categorical variables with a lot of levels	

103

Check Your Knowledge

Your Thoughts?

1. Consider the following Training Data Set:
   - Apply the Naïve Bayesian Classifier to this data set and compute the probability score for $P(y = 1|X)$ for $X = (1,0,0)$

   Show your work

Training Data Set

X1	X2	X3	Y
1	1	1	0
1	1	0	0
0	0	0	0
0	1	0	1
1	0	1	1
0	1	1	1

2. List some prominent use cases of the Naïve Bayesian Classifier.
3. What gives the Naïve Bayesian Classifier the advantage of being computationally inexpensive?
4. Why should we use log-likelihoods rather than pure probability values in the Naïve Bayesian Classifier?

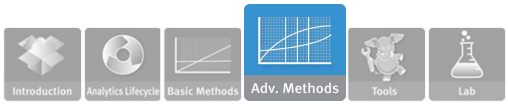
104

Check Your Knowledge (Continued)

5. What is a confusion matrix and how it is used to evaluate the effectiveness of the model?
6. Consider the following data set with two input features temperature and season
   - What is the Naïve Bayesian assumption?
   - Is the Naïve Bayesian assumption satisfied for this problem?

Temperature	Season	Electricty Usage
-10 to 50 F	Winter	High
50 to 70 F	Winter	Low
70 to 85 F	Summer	Low
85 to 110 F	Summer	High

105



Module 4: Advanced Analytics – Theory and Methods

Lesson 5: Naïve Bayesian Classifiers - Summary

During this lesson the following topics were covered:

- Naïve Bayesian Classifier
- Theoretical foundations of the classifier
- Use cases
- Evaluating the effectiveness of the classifier
- The Reasons to Choose (+) and Cautions (-) with the use of the classifier

106

Lab Exercise 8: Naïve Bayesian Classifier

This Lab is designed to investigate and practice the Naïve Bayesian Classifier analytic technique.

After completing the tasks in this lab you should be able to:

- Use R functions for Naïve Bayesian Classification
- Apply the requirements for generating appropriate training data
- Validate the effectiveness of the Naïve Bayesian Classifier with the big data

107

Lab Exercise 8: Naïve Bayesian Classifier Part1 - Workflow

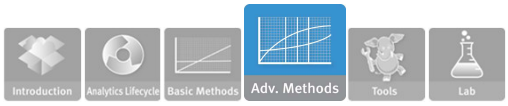
1. Set working directory and review training and test data
2. Install and load library “e1071”
3. Read in and review data
4. Build the Naïve Bayesian classifier model from first principles
5. Predict the results
6. Use the naiveBayes function
7. Predict the Outcome of “Enrolls” with the test data
8. Use the Laplace smoothing

108

Lab Exercise 8: Naïve Bayesian Classifier Part2 - Workflow

1. Define the problem *(Translating to an Analytics Question)*
2. Open the ODBC connection
3. Build the training dataset and the test dataset from the database
4. Extract the first 10000 records for the training data set and the remaining 10 for the test
5. Execute the NB classifier
6. Validate the effectiveness of the NB classifier with a confusion matrix
7. Execute NB classifier with MADlib function calls within the database

109



Module 4: Advanced Analytics – Theory and Methods

Lesson 6: Decision Trees

During this lesson the following topics are covered:

- Overview of Decision Tree classifier
- General algorithm for Decision Trees
- Decision Tree use cases
- Entropy, Information gain
- Reasons to Choose (+) and Cautions (-) of Decision Tree classifier
- Classifier methods and conditions in which they are best suited

110

Decision Tree

Decision Trees are a flexible method very commonly deployed in data mining applications. In this lesson we will focus on Decision Trees used for **classification problems**.

There are two types of trees; Classification Trees and Regression (or Prediction) Trees

- Classification Trees – are used to segment observations into more homogenous groups (assign class labels). They usually apply to outcomes that are binary or categorical in nature.

- **Regression Trees** – are variations of regression and what is returned in each node is the average value at each node (type of a step function with which the average value can be computed). Regression trees can be applied to outcomes that are continuous (like account spend or personal income).

111

Decision Tree Classifier - What is it?

- Used for classification:
  - Returns probability scores of class membership
    - Well-calibrated, like logistic regression
    - Assigns label based on highest scoring class
    - Some Decision Tree algorithms return simply the most likely class
  - Regression Trees: a variation for regression
    - Returns average value at every node
    - Predictions can be discontinuous at the decision boundaries
- Input variables can be continuous or discrete
- Output:
  - **A tree** that describes the decision flow.
  - **Leaf nodes** return either a probability score, or simply a classification.
  - Trees can be converted to a set of "**decision rules**"
    - "IF income < $50,000 AND mortgage_amt > $100K THEN default=T with 75% probability"

112

Decision Tree

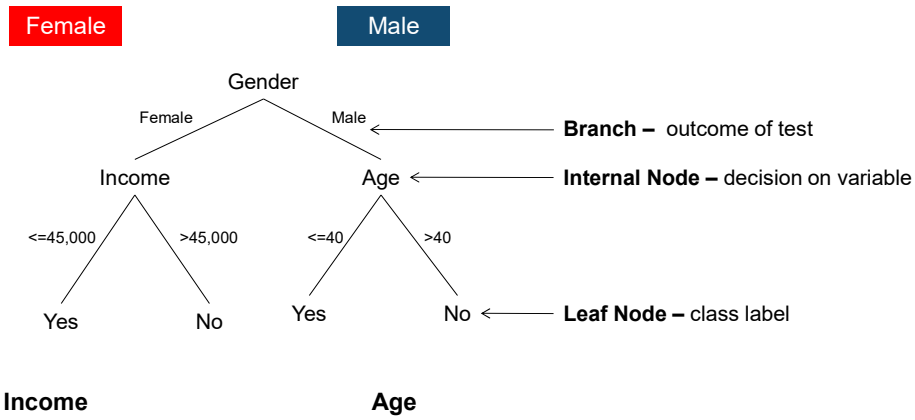
Decision Trees are a popular method because they can be applied to a variety of situations.

The rules of classification are very straight forward and the results can easily be presented visually.

Additionally, because the end result is **a series of logical "if-then" statements**, there is no underlying assumption of a **linear (or non-linear) relationship between the predictor variables and the dependent variable.**

113

Decision Tree – Example of Visual Structure



114

Decision Tree Classifier - Use Cases

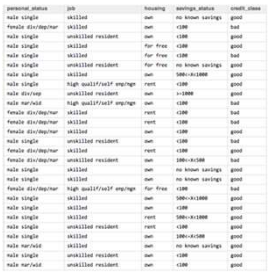
- When a series of questions (yes/no) are answered to arrive at a classification
  - Biological species classification
  - Checklist of symptoms during a doctor's evaluation of a patient
- When "if-then" conditions are preferred to linear models.
  - Customer segmentation to predict response rates
  - Financial decisions such as loan approval
  - Fraud detection
- Short Decision Trees are the most popular "weak learner" in ensemble learning techniques

115

Example: The Credit Prediction Problem

Class labels: {good, bad}
- P(good) = 0.7
- P(bad) = 0.3

Conditional Probabilities
- P(own|bad) = 0.62
- P(own|good) = 0.75
- P(rent|bad) = 0.23
- P(rent|good) = 0.14
- ... and so on



116

Example: The Credit Prediction Problem

We will use the same example we used in the previous lesson with Naïve Bayesian classifier.

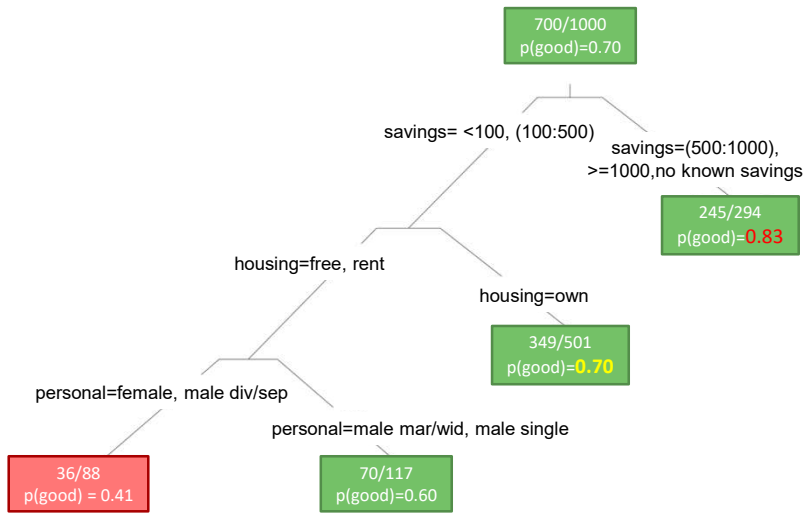
For the people with good credit and we start at the top of the tree the probability is 70% (700 out of 1000 people have good credit). The process has decided that we are going to split how much is in the savings account into two groups.

One group with savings less than $100 or between $100 to $ 500.

The second group is the rest of the population which has savings of $500 to $1000 or greater than $1000 or no known savings.

117

Example: The Credit Prediction Problem



118

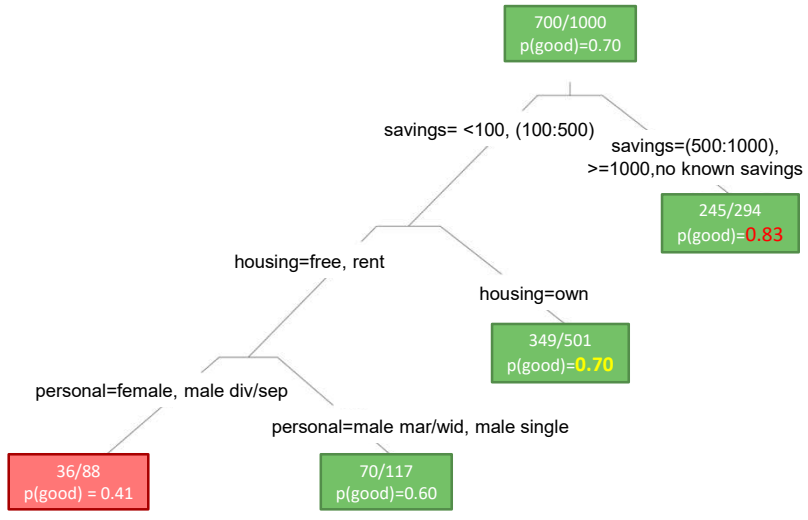
Example: The Credit Prediction Problem

We compute the probability of good credit at the second node and we find in the second savings category 245 out of 294 have **good credit** and the probability at this node is **83%**.

Looking at the other node (Savings <100 or Savings 100:500) we look into housing. We split this node into Housing (free,rent) as one group and Housing (own) as the other. Computing probability of good credit at housing (own) node we see that 349 out of 501 people have good credit, a **70%** probability.

119

Example: The Credit Prediction Problem



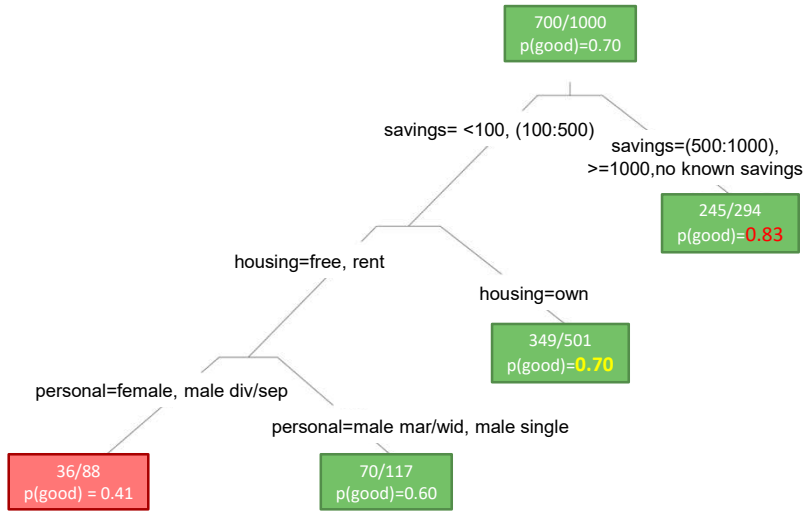
120

Example: The Credit Prediction Problem

Traversing down the housing (free, rent) node we split now on the variable known as personal. The two groups are Personal (female, male divorced/ separated) and Personal (male,married/widowed,male_single). In the node on the right, the probability of good credit is 0.6; in the node on the left, the probability of good credit is 41% (which is less than 50%, so we have shaded this box red).

121

Example: The Credit Prediction Problem



122

Example: The Credit Prediction Problem

We can see that for this case, we might want to work with the probabilities, rather than the class labels; this tree would only label 88 rows (out of 1000) of the training set as "bad", which is far less than the 30% "bad" rate of the training set, and of those cases labeled "bad", only 59% of them would truly be bad.

Tuning the splitting parameters, or using a **random forest** or other ensemble technique (more on that later) might improve the performance.

123

Example: The Credit Prediction Problem

Decision Trees are greedy algorithms. They take decisions based on what is available at that moment and once a bad decision is taken it is propagated all the way down.

An ensemble technique may randomize the splitting (or even randomize data) and come up with multiple tree structures. It then assigns labels by looking at the average of the nodes in all the trees and assigns class labels or probability values.

124

General Algorithm

- To construct tree T from training set S
  - If all examples in S belong to some class in C, or S is sufficiently "pure", then make a leaf labeled C.
  - Otherwise:
    - select the “most informative” attribute A
    - partition S according to A’s values
    - recursively construct sub-trees T1, T2, ..., for the subsets of S
- The details vary according to the specific algorithm – CART, ID3, C4.5 – but the general idea is the same

125

Step 1: Pick the Most “Informative" Attribute

- Entropy-based methods are one common way

$$H = -\sum_{c} p(c) \log_2 p(c)$$

- H = 0 if p(c) = 0 or 1 for any class
  - So for binary classification, H=0 is a "pure" node
- H is maximum when all classes are equally probable
  - For binary classification, H=1 when classes are 50/50

126

Step 1: Pick the most "informative" attribute (Continued)

- First, we need to get the base entropy of the data

$$H_{credit} = -(0.7\log_2(0.7) + 0.3\log_2(0.3)) = 0.88$$

127

Step 1: Pick the Most "Informative" Attribute (Continued) Conditional Entropy

$$H_{attr} = -\sum_v p(v) \sum_c p(c|v)\log_2 p(c|v)$$

- The weighted sum of the class entropies for each value of the attribute
- In English: attribute values (home owner vs. renter) give more information about class membership
  - "Home owners are more likely to have good credit than renters"
- Conditional entropy should be lower than unconditioned entropy

128

Conditional Entropy Example

Let's compute the conditional entropy of credit class conditioned on housing status. In the top row of the table are the probabilities of each value. In the next two rows are the probabilities of the class labels conditioned on the housing value.

Note that each term inside parentheses is the entropy of the class labels within a single housing value.

The conditional entropy is still fairly high; but it is a little less than the unconditioned entropy.

	for free	own	rent
P(housing)	0.108	0.713	0.179
P(bad | housing)	0.407	0.261	0.391
p(good | housing)	0.592	0.739	0.601

$$\begin{aligned} H_{(housing|credit)} &= -[0.108 * (0.407 \log_2(0.407) + 0.592 \log_2(0.592)) \\ &\quad + 0.713 * (0.261 \log_2(0.261) + 0.739 \log_2(0.739)) \\ &\quad + 0.179 * (0.391 \log_2(0.391) + 0.601 \log_2(0.601))] \\ &= 0.868 \end{aligned}$$

129

Step 1: Pick the Most "Informative" Attribute (Continued) Information Gain

$$\text{InfoGain}_{attr} = H - Hattr$$

- The information that you gain, by knowing the value of an attribute
- So the "most informative" attribute is the attribute with the highest InfoGain

130

Back to the Credit Prediction Example

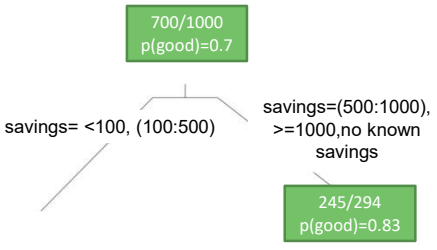
$$\begin{aligned}\text{InfoGain}_{credit} &= H_{credit} - H_{housing|credit} \\ &= 0.88 - 0.86 \\ &\approx 0.013\end{aligned}$$

Attribute	InfoGain
job	0.001
housing	0.013
personal_status	0.006
savings_status	***0.028***

131

Step 2 & 3: Partition on the Selected Variable

- Step 2: Find the partition with the highest InfoGain
  - In our example the selected partition has InfoGain = 0.028
- Step 3: At each resulting node, repeat Steps 1 and 2
  - until node is "pure enough"
- Pure nodes => no information gain by splitting on other attributes



132

Diagnostics

- Hold-out data
- ROC/AUC
- Confusion Matrix
- FPR/FNR, Precision/Recall
- Do the splits (or the "rules") make sense?
  - What does the domain expert say?
- How deep is the tree?
  - Too many layers are prone to over-fit
- Do you get nodes with very few members?
  - Over-fit

133

Decision Tree Classifier - Reasons to Choose (+) & Cautions (-)

Reasons to Choose (+)	Cautions (-)
Takes any input type (numeric, categorical) In principle, can handle categorical variables with many distinct values (ZIP code)	Decision surfaces can only be axis-aligned
Robust with redundant variables, correlated variables	Tree structure is sensitive to small changes in the training data
Naturally handles variable interaction	A "deep" tree is probably over-fit Because each split reduces the training data for subsequent splits
Handles variables that have non-linear effect on outcome	Not good for outcomes that are dependent on many variables Related to over-fit problem, above
Computationally efficient to build	Doesn't naturally handle missing values; However most implementations include a method for dealing with this
Easy to score data	In practice, decision rules can be fairly complex
Many algorithms can return a measure of variable importance	
In principle, decision rules are easy to understand	

134

Decision Tree Classifier - Reasons to Choose (+) & Cautions (-) (Continued)

Reasons to Choose (+)	Cautions (-)
Takes any input type (numeric, categorical) In principle, can handle categorical variables with many distinct values (ZIP code)	Decision surfaces can only be axis-aligned
Robust with redundant variables, correlated variables	Tree structure is sensitive to small changes in the training data
Naturally handles variable interaction	A "deep" tree is probably over-fit because each split reduces the training data for subsequent splits
Handles variables that have non-linear effect on outcome	Not good for outcomes that are dependent on many variables (Related to over-fit problem, above)
Computationally efficient to build	Doesn't naturally handle missing values; However most implementations include a method for dealing with this
Easy to score data	In practice, decision rules can be fairly complex
Many algorithms can return a measure of variable importance	
In principle, decision rules are easy to understand	

135

Which Classifier Should I Try?

Typical Questions	Recommended Method
Do I want class probabilities, rather than just class labels?	Logistic regression Decision Tree
Do I want insight into how the variables affect the model?	Logistic regression Decision Tree
Is the problem high-dimensional?	Naïve Bayes
Do I suspect some of the inputs are correlated?	Decision Tree Logistic Regression
Do I suspect some of the inputs are irrelevant?	Decision Tree Naïve Bayes
Are there categorical variables with a large number of levels?	Naïve Bayes Decision Tree
Are there mixed variable types?	Decision Tree Logistic Regression
Is there non-linear data or discontinuities in the inputs that will affect the outputs?	Decision Tree

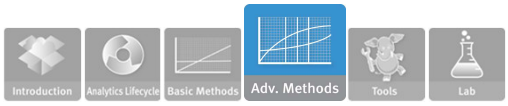
136

Check Your Knowledge

Your Thoughts?

1. How do you define information gain?
2. For what conditions is the value of entropy at a maximum and when is it at a minimum?
3. List three use cases of Decision Trees.
4. What are weak learners and how are they used in ensemble methods?
5. Why do we end up with an over fitted model with deep trees and in data sets when we have outcomes that are dependent on many variables?
6. What classification method would you recommend for the following cases:
   - High dimensional data
   - Data in which outputs are affected by non-linearity and discontinuity in the inputs

137



Module 4: Advanced Analytics – Theory and Methods

Lesson 6: Decision Trees - Summary

During this lesson the following topics were covered:

- Overview of Decision Tree classifier
- General algorithm for Decision Trees
- Decision Tree use cases
- Entropy, Information gain
- Reasons to Choose (+) and Cautions (-) of Decision Tree classifier
- Classifier methods and conditions in which they are best suited

138

Lab Exercise 9: Decision Trees

This lab is designed to investigate and practice Decision Tree models covered in the course work.

After completing the tasks in this lab you should be able to:

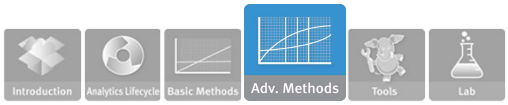
- Use R functions for Decision Tree models
- Predict the outcome of an attribute based on the model

139

Lab Exercise 9: Decision Trees - Workflow

1. Set the Working Directory
2. Read in the Data
3. Build the Decision Tree
4. Plot the Decision Tree
5. Prepare Data to Test the Fitted Model
6. Predict a Decision from the Fitted Model

140



Module 4: Advanced Analytics – Theory and Methods

Lesson 7: Time Series Analysis

During this lesson the following topics are covered:

- Time Series Analysis and its applications in forecasting
- ARMA and ARIMA Models
- Implementing the Box-Jenkins Methodology using R
- Reasons to Choose (+) and Cautions (-) with Time Series Analysis

141

Time Series Analysis

Businesses perform sales forecasting to look ahead in order to plan their investments, launch new products, decide when to close or withdraw products, etc.

The sales forecasting process is a critical one for most businesses. Part of the sales forecasting process is to examine the past. How well did we do in the last few months or what were our sales in the same time period for the last few years?

Time Series Analysis provides a scientific methodology for sales forecasting. **Time Series Analysis** is the analysis of sequential data across equally spaced units of time. Time Series is a basic research methodology in which data for one or more variables are collected for many observations at different time periods.

142

Time Series Analysis

The main objectives in Time Series Analysis are:

- To understand the underlying structure of the time series by breaking it down to its components.
- Fit a mathematical model and then proceed to forecast the future

The time periods are usually regularly spaced and the observations may be either univariate or multivariate. **Univariate** time series are those where only one variable is measured over time, whereas multivariate time series are those, where multiple variables are measured simultaneously. The internal structure of the data may specify a trend, seasonality, cycles or random.

143

Time Series Analysis

- Time Series: Ordered sequence of equally spaced values over time
- Time Series Analysis: Accounts for the **internal structure** of observations taken over time
  - Trend – long term movement in a time series
  - Seasonality – dependent on the time of the year
  - Cycles – non-seasonal nature
  - Random – or chaotic values left over when other components of the series have been accounted for.
- Goals
  - To identify the internal structure of the time series
  - To forecast future events
    - Example: Based on sales history, what will next December sales be?
- **Method: Box-Jenkins (ARMA – Auto Regressive Moving Averages)**

144

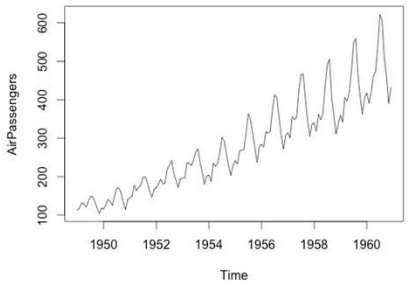
Time Series Analysis (Continued)

- Time Series: Ordered sequence of equally spaced values over time
- Time Series Analysis: Accounts for the **internal structure** of observations taken over time
  - Trend
  - Seasonality
  - Cycles
  - Random
- Goals
  - To identify the internal structure of the time series
  - To forecast future events
    - Example: Based on sales history, what will next December sales be?
- **Method: Box-Jenkins (ARMA)**

145

Box-Jenkins Method: What is it?

- Models historical behavior to forecast the future



- Applies ARMA (Autoregressive Moving Averages)
  - Input: Time Series
    - *Accounting for Trends and Seasonality components*
  - Output: Expected future value of the time series

146

Use Cases

Forecast:

- Next month's sales
- Tomorrow's stock price
- Hourly power demand

- Economics/Finance
- Sociology
- Epidemiology:
  - SARS/MERS/COVID-19

147

Modeling a Time Series

- Let's model the time series as

$$Y_t = T_t + S_t + R_t, \quad t=1,\ldots,n.$$

- T_t: Trend term
  - Air travel steadily increased over the last few years
- S_t: The seasonal term
  - Air travel fluctuates in a regular pattern over the course of a year
- R_t: Random component
  - To be modeled with ARMA

148

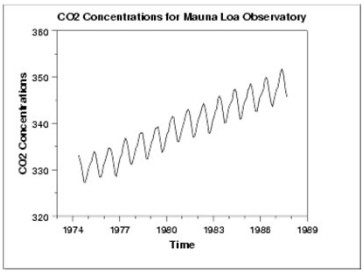
Stationary Sequences

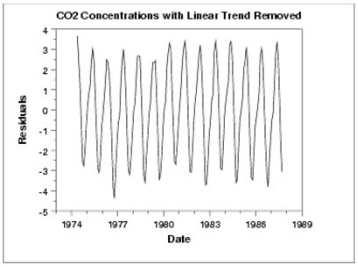
- Box-Jenkins methodology assumes the random component is a *stationary sequence*
  - Constant mean
  - Constant variance
  - Autocorrelation does not change over time
    - Constant correlation of a variable with itself at different times
- In practice, to obtain a stationary sequence, the data must be:
  - De-trended
  - Seasonally adjusted

149

De-trending

- In this example, we see a linear trend, so we fit a linear model
  - $T_t = m \cdot t + b$
- The de-trended series is then
  - $Y^1_t = Y_t - T_t$
- In some cases, may have to fit a non-linear model
  - Quadratic
  - Exponential



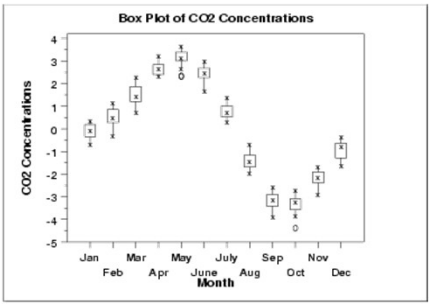


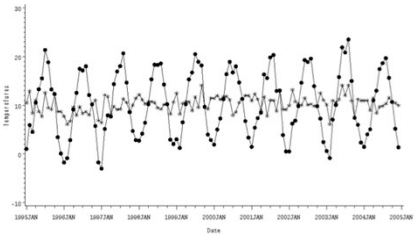
150

Seasonal Adjustment

- Plotting the de-trended series identifies seasons
  - For CO2 concentration, we can model the period as being a year, with variation at the month level
- Simple ad-hoc adjustment: take several years of data, calculate the average value for each month, and subtract that from Y^1_t

$$Y^2_t = Y^1_t - S_t$$





151

ARMA(p, q) Model – Auto Regressive Moving Averages

$$Y_t = \delta + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q}$$

- The simplest Box-Jenkins Model
  - Y_t is de-trended and seasonally adjusted
- Combination of two process models
  - **Autoregressive**: Y_t is a linear combination of its last *p* values
  - **Moving average**: Y_t is a constant value plus the effects of a dampened white noise process over the last *q* time values (lags)

152

ARIMA(p, d, q) Model - Auto Regressive Integrated Moving Averages

- ARIMA adds a differencing term, *d*, to the ARMA model
  - Autoregressive Integrated Moving Average
  - Includes the de-trending as part of the model
    - linear trend can be removed by *d*=1
    - quadratic trend by *d*=2
    - and so on for higher order trends
- The general non-seasonal model is known as ARIMA (*p, d, q*):
  - *p* is the number of autoregressive terms
  - *d* is the number of differences
  - *q* is the number of moving average terms

153

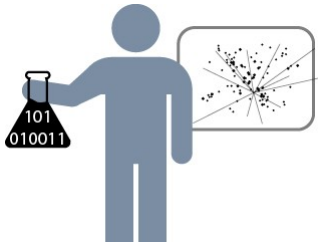
ACF & PACF

- Auto Correlation Function (ACF)
  - Correlation of the values of the time series with itself
  - Autocorrelation "carries over"
  - Helps to determine the order, q, of a MA model
    - Where does ACF go to zero?
- Partial Auto Correlation Function (PACF)
  - An autocorrelation calculated after removing the linear dependence of the previous terms
  - Helps to determine the order, *p*, of an AR model
    - Where does PACF go to zero?

154

Model Selection

- Based on the data, the Data Scientist selects *p, d* and *q*
  - An "art form" that requires domain knowledge, modeling experience, and a few iterations
  - Use a simple model when possible
    - AR model (q = 0)
    - MA model (p = 0)
- Multiple models need to be built and compared
  - Using ACF and PACF



155

Time Series Analysis - Reasons to Choose (+) & Cautions (-)

Reasons to Choose (+)	Cautions (-)
Minimal data collection Only have to collect the series itself Do not need to input drivers	No meaningful drivers: prediction based only on past performance No explanatory value Can't do "what-if" scenarios Can't stress test
Designed to handle the inherent autocorrelation of lagged time series	It's an "art form" to select appropriate parameters
Accounts for trends and seasonality	Only suitable for short term predictions

156

Time Series Analysis with R

Important R functions and commands we will be using are listed here.

- The function "*ts*" is used to create time series objects
  - **mydata<- ts(mydata,start=c(1999,1),frequency=12)**
- Visualize data
  - **plot(mydata)**
- De-trend using differencing
  - **diff(mydata)**
- Examine ACF and PACF
  - **acf(mydata)**: It computes and plots estimates of the autocorrelations
  - **pacf(mydata)**: It computes and plots estimates of the partial autocorrelations

157

Other Useful R Functions in Time Series Analysis

- **ar()**: Fit an autoregressive time series model to the data
- **arima()**: Fit an ARIMA model
- **predict()**: Makes predictions
  - *"predict"* is a generic function for predictions from the results of various model fitting functions. The function invokes particular methods which depend on the *class* of the first argument
- **arima.sim()**: Simulate a time series from an ARIMA model
- **decompose()**: Decompose a time series into seasonal, trend and irregular components using moving averages
  - Deals with additive or multiplicative seasonal component
- **stl():** Decompose a time series into seasonal, trend and irregular components using loess

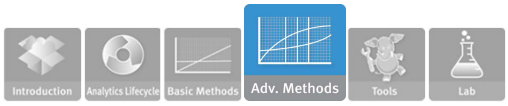
158

Check Your Knowledge

Your Thoughts?

1. What is a time series and what are the key components of a time series?
2. How do we "de-trend" a time series data?
3. What makes data stationary?
4. How is seasonality removed from the data?
5. What are the modeling parameters in ARIMA?
6. How do you use ACF and PACF to determine the "stationarity" of time series data?

159



Module 4: Advanced Analytics – Theory and Methods

Lesson 7: Time Series Analysis - Summary

During this lesson the following topics were covered:

- Time Series Analysis and its applications in forecasting
- ARMA and ARIMA Models
- Implementing the Box-Jenkins Methodology using R
- Reasons to Choose (+) and Cautions (-) with Time Series Analysis

160

Lab Exercise 10: Time Series Analysis

This Lab is designed to investigate and practice Time Series Analysis with ARIMA models (Box-Jenkins-methodology).

After completing the tasks in this lab you should be able to:

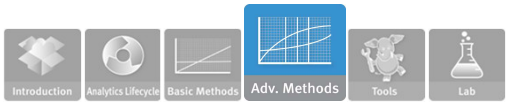
- Use R functions for ARIMA models
- Apply the requirements for generating appropriate training data
- Validate the effectiveness of the ARIMA models

161

Lab Exercise 10: Time Series Analysis - Workflow

1. Set the Working Directory
2. Open Connection to Database
3. Get Data from the Database
4. Import the Table
5. Review, Update, and Prepare DataFrame "msales" File for ARIMA Modeling
6. Convert "sales" into Time Series Object
7. Plot the Time Series
8. Analyze the ACF and PACF
9. Difference the Data to Make it Stationary
10. Plot ACF and PACF for the Differenced Data
11. Fit the ARIMA Model
12. Generate Predictions
13. Compare Predicted Values with Actual Values

162



Module 4: Advanced Analytics – Theory and Methods

Lesson 8: Text Analysis

During this lesson the following topics are covered:

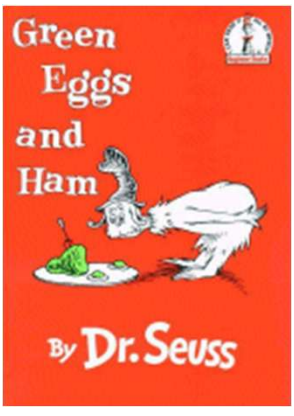
- Challenges with text analysis
- Key tasks in text analysis
- Definition of terms used in text analysis
  - Term frequency, inverse document frequency
- Representation and features of documents and corpus
- Use of regular expressions in parsing text
- Metrics used to measure the quality of search results
  - Relevance with tf-idf, precision and recall

163

Text Analysis

Encompasses the processing and representation of text for analysis and learning tasks

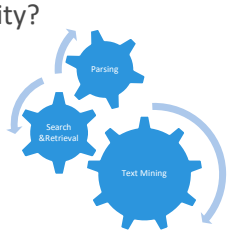
- **High-dimensionality**
  - Every distinct term is a dimension
  - *Green Eggs and Ham*: A 50-D problem!
- **Data is Un-structured**



164

Text Analysis – Problem-solving Tasks

- Parsing
  - Impose a structure on the unstructured/semi-structured text for downstream analysis
- Search/Retrieval
  - Which documents have this word or phrase?
  - Which documents are about this topic or this entity?
- Text-mining
  - "Understand" the content
  - Clustering, classification
- Tasks are not an ordered list
  - Does not represent process
  - Set of tasks used appropriately depending on the problem addressed



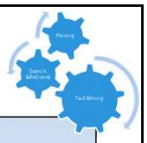
165

Example: Brand Management

- Acme currently makes two products
  - bPhone
  - bEbook
- They have lots of competition. They want to maintain their reputation for excellent products and keep their sales high.
- What is the buzz on Acme?
  - Search for mentions of Acme products
    - Twitter, Facebook, Review Sites, etc.
  - What do people say?
    - Positive or negative?
    - What do people think is good or bad about the products?

166

Buzz Tracking: The Process



1. Monitor social networks, review sites for mentions of our products.	**Parse** the data feeds to get actual content. Find and filter the raw text for product names (Use **Regular Expression**).
2. Collect the reviews.	Extract the relevant raw text. Convert the raw text into a suitable **document representation**. **Index** into our review **corpus**.
3. Sort the reviews by product.	**Classification** (or **"Topic Tagging"**)
4. Are they good reviews or bad reviews? We can keep a simple count here, for trend analysis.	**Classification** (sentiment analysis)
5. Marketing calls up and reads selected reviews in full, for greater insight.	**Search/Information Retrieval**.

167

Parsing the Feeds

1. Monitor social networks, review sites for mentions of our products

- Impose structure on semi-structured data.
- We need to know where to look for what we are looking for.

```
<channel>
<title>All about Phones</title>
<description>My Phone Review Site</description>
<link>http://www.phones.com/link.htm</link>

<item>
<title>bPhone: The best!</title>
<description>I love LOVE my bPhone!</description>
<link>http://www.phones.com/link.htm</link>
<guid isPermaLink="false"> 1102345</guid>
<pubDate>Tue, 29 Aug 2011 09:00:00 -0400</pubDate>
</item>

</channel>
```


168

Regular Expressions

1. Monitor social networks, review sites for mentions of our products

- Regular Expressions (regexp) are a means for finding words, strings or particular patterns in text.
- A match is a Boolean response. The basic use is to ask "does this regexp match this string?"

regexp	matches	Note
b[P|p]hone	bPhone, bphone	Pipe "|" means "or"
bEbo*k	bEbk, bEbok, bEbook, bEboook …	"*" matches 0 or more repetitions of the preceding letter
^I love	A line starting with "I love"	"^" means start of a string
Acme$	A line ending with "Acme"	"$" means the end of a string

169

Extract and Represent Text

2. Collect the reviews

Document Representation:
A structure for analysis

- **"Bag of words"**
  - common representation
  - A vector with one dimension for every unique term in space
    - **term-frequency (tf)**: number times a term occurs
  - Good for basic search, classification
- Reduce Dimensionality
  - Term Space – not ALL terms
    - no stop words: "the", "a"
    - often no pronouns
  - Stemming
    - "phone" = "phones"

"I love LOVE my bPhone!"

Convert this to a vector in the term space:

acme	0
bebook	0
bPhone	1
fantastic	0
love	2
slow	0
terrible	0
terrific	0

170

Document Representation - Other Features

2. Collect the reviews

- Feature:
  - Anything about the document that is used for search or analysis.
- Title
- Keywords or tags
- Date information
- Source information
- Named entities

171

Representing a Corpus (Collection of Documents)

2. Collect the reviews

- Reverse index
  - For every possible feature, a list of all the documents that contain that feature
- Corpus metrics
  - Volume
  - Corpus-wide term frequencies
  - Inverse Document Frequency (IDF)
    - more on this later
- Challenge: a Corpus is dynamic
  - Index, metrics must be updated continuously

172

Text Classification (I) - "Topic Tagging"

3. Sort the Reviews by Product

Not as straightforward as it seems

"The bPhone-5X has coverage everywhere. It's much less flaky than my old bPhone-4G."

"While I love Acme's bPhone series, I've been quite disappointed by the bEbook. The text is illegible, and it makes even my old Newton look blazingly fast."

173

"Topic Tagging"

3. Sort the Reviews by Product

Judicious choice of features

- Product mentioned in title?
- Tweet, or review?
- Term frequency
- Canonicalize abbreviations
  - "5X" = "bPhone-5X"

174

Text Classification (II) Sentiment Analysis

4. Are they good reviews or bad reviews?

- Naïve Bayes is a good first attempt
- But you need tagged training data!
  - The major bottleneck in text classification
- What to do?
  - Hand-tagging
  - Clues from review sites
    - thumbs-up or down, # of stars
  - Cluster documents, then label the clusters

175

Search and Information Retrieval

5. Marketing calls up and reads selected reviews in full, for greater insight.

- Marketing calls up documents with *queries*:
  - Collection of search terms
    - "bPhone battery life"
  - Can also be represented as "bag of words"
  - Possibly restricted by other attributes
    - within the last month
    - from this review site

176

Quality of Search Results

5. Marketing calls up and reads selected reviews in full, for greater insight.

- Relevance
  - Is this document what I wanted?
  - Used to rank search results
- Precision
  - What % of documents in the result are relevant?
- Recall
  - Of all the relevant documents in the corpus, what % were returned to me?

177

Computing Relevance (Term Frequency)

5. Marketing calls up and reads selected reviews in full, for greater insight.

- Assign each term in a document a weight for that term.
- The weight of a term *t* in a document *d* is a function of the number of times *t* appears in *d*.
  - The weight can be simply set to the number of occurrences of *t* in *d* :

$$tf\,(t, d) = count\,(t, d)$$

  - The term frequency may optionally be normalized.

178

Inverse Document Frequency (idf)

5. Marketing calls up and reads selected reviews in full, for greater insight.

$$idf(t) = log\ [N/df(t)]$$

- *N*: Number of documents in the corpus
- *df*(t): Number of documents in the corpus that contain a term *t*

- Measures term uniqueness in corpus
  - "phone" vs. "brick"
- Indicates the importance of the term
  - Search (relevance)
  - Classification (discriminatory power)

179

TF-IDF and Modified Retrieval Algorithm

5. Marketing calls up and reads selected reviews in full, for greater insight.

- Term frequency – inverse document frequency (tf-idf or tfidf) of term t in document d:

$$tfidf(t, d) = tf(t, d) * idf(t)$$

query: *brick, phone*

- Document with "brick" a few times more relevant than document with "phone" many times
- Measure of Relevance with tf-idf
- Call up all the documents that have any of the terms from the query, and sum up the tf-idf of each term:

$$\text{Relevance}(d) = \sum_{i \in [1,n]} tfidf(t_i, d)$$

180

Other Relevance Metrics

5. Marketing calls up and reads selected reviews in full, for greater insight.

- "Authoritativeness" of source
  - PageRank is an example of this
- Recency of document
- How often the document has been retrieved by other users

181

Effectiveness of Search and Retrieval

- Relevance metric
  - important for precision, user experience
- Effective crawl, extraction, indexing
  - important for recall (and precision)
  - more important, often, than retrieval algorithm
- MapReduce
  - Reverse index, corpus term frequencies, idf

182

Natural Language Processing

- Unstructured text mining means extracting "features"
  - Features are structured meta-data representing the document
  - Goal: "vectorize" the documents
- After vectorization, apply advanced machine learning techniques
  - Clustering
  - Classification
    - Decision Trees
    - Naïve Bayesian Classifier
  - Scoring
    - Once models have been built, use them to automatically categorize incoming documents

183

Example: UFOs Attack

July 15th, 2010. Raytown, Missouri

When I fist noticed it, I wanted to freak out. There it was an object floating in on a direct path, It didn't move side to side or volley up and down. It moved as if though it had a mission or purpose. I was nervous, and scared, So afraid in fact that I could feel my knees buckling. I guess because I didn't know what to expect and I wanted to act non aggressive. I though that I was either going to be taken, blasted into nothing, or…

Q: What is the witness describing?

A: An encounter with a UFO.

Q: What is the emotional state of the witness?

A: Frightened, ready to flee.

Source: http://www.infochimps.com/datasets/60000-documented-ufo-sightings-with-text-descriptions-and-metada

184

Example: UFOs Attack

If we really are on the cusp of a major alien invasion, eyewitness testimony is the key to our survival as a species.

*Strangely, the computer finds this account **unreliable!***

*When I **fist** noticed it, I wanted to freak out ject floatin in on a direct path, It **didn't** move side to side or volley up n. It moved as if though it had a mission or purpose. I was and scared, **So afraid in fact** that I could feel my knees buckling. I guess because I didn't kno d I wanted t h that I was either going to be **taken**, blasted into nothing, or…*

Machine error

Typo

Turn of phrase

Ambiguous meaning

"UFO" keyword missing

Source: http://www.infochimps.com/datasets/60000-documented-ufo-sightings-with-text-descriptions-and-metada

185

Example: UFOs Attack

Investigators need to…

Search	*for keywords and phrases, but your topic may be very complicated or keywords may be misspelled within the document*
Manage	*document meta-data like time, location and author. Later retrieval may be key to identifying this meta-data early, and the document may be amenable to structure.*
Understand	*content via sentiment analysis, custom dictionaries, natural language processing, clustering, classification and good ol' domain expertise.*

…with computer-aided text mining

186

Challenges - Text Analysis

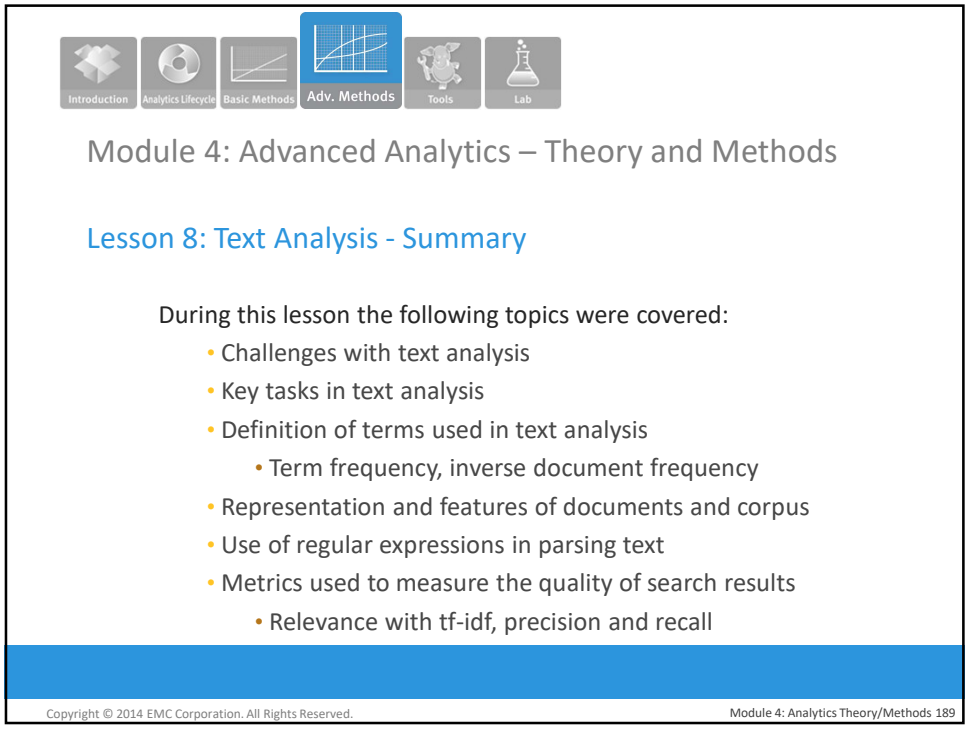
1. Finding the right structure for your unstructured data
2. Very high dimensionality
3. Thinking about your problem the right way

187

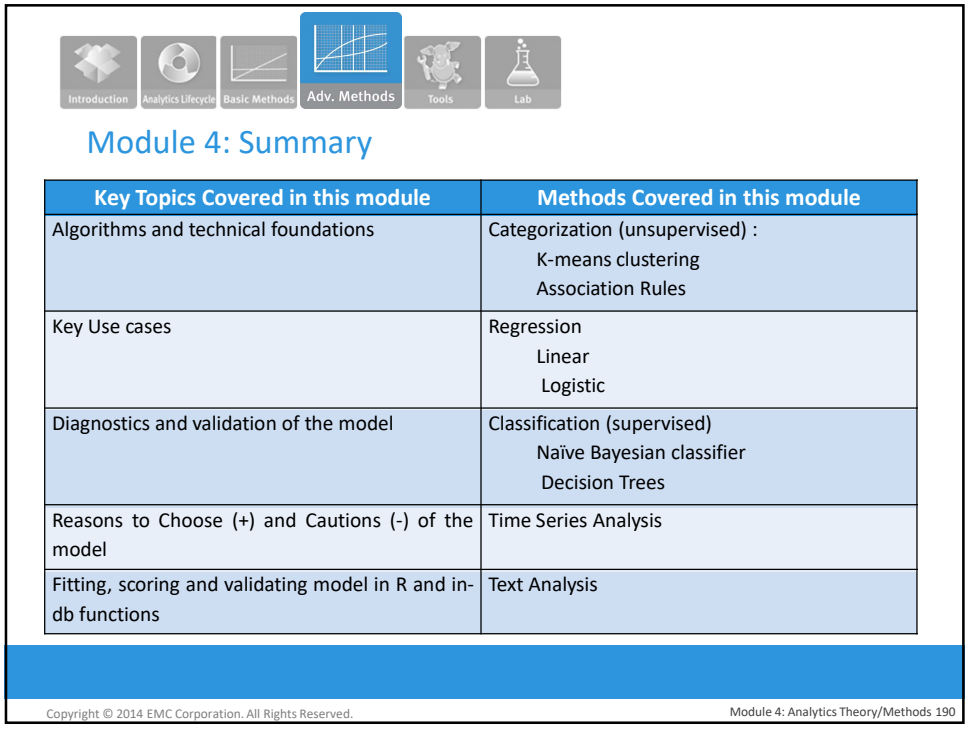
Check Your Knowledge

1. What are the two major challenges in the problem of text analysis?
2. What is a reverse index?
3. Why is the corpus metrics dynamic. Provide an example and a scenario that explains the dynamism of the corpus metrics.
4. How does tf-idf enhance the relevance of a search result?
5. List and discuss a few methods that are deployed in text analysis to reduce the dimensions.

188


Introduction
Analytics Lifecycle
Basic Methods
Adv. Methods
Tools
Lab
Module 4: Advanced Analytics – Theory and Methods
Lesson 8: Text Analysis - Summary
During this lesson the following topics were covered:
• Challenges with text analysis
• Key tasks in text analysis
• Definition of terms used in text analysis
• Term frequency, inverse document frequency
• Representation and features of documents and corpus
• Use of regular expressions in parsing text
• Metrics used to measure the quality of search results
• Relevance with tf-idf, precision and recall
Copyright © 2014 EMC Corporation. All Rights Reserved.
Module 4: Analytics Theory/Methods 189

189


Introduction
Analytics Lifecycle
Basic Methods
Adv. Methods
Tools
Lab
Module 4: Summary
Key Topics Covered in this module
Algorithms and technical foundations
Key Use cases
Diagnostics and validation of the model
Reasons to Choose (+) and Cautions (-) of the model
Fitting, scoring and validating model in R and in-db functions
Methods Covered in this module
Categorization (unsupervised) :
K-means clustering
Association Rules
Regression
Linear
Logistic
Classification (supervised)
Naïve Bayesian classifier
Decision Trees
Time Series Analysis
Text Analysis
Copyright © 2014 EMC Corporation. All Rights Reserved.
Module 4: Analytics Theory/Methods 190

190