Week 1: Introduction to Machine Learning

1. R as an Analytics Language Revisited

* R review
* Data Structure
* Data Semantics Numerical vs non-numerical
  + A numerical variable is a variable where the measurement or number has a numerical meaning. For example, total Saint Louis rainfall measured in inches is a numerical value, your heart rate is a numerical value, number of electrical unit consumed in an hour is a numerical value.
  + A nominal variable is a type of variable that is used to name, label or categorize particular attributes that are being measured. It takes qualitative values representing different categories, and there is no intrinsic ordering of these categories. We can code nominal variables with numbers, but the order is arbitrary and arithmetic operations cannot be performed on the numbers. This is the case when a person’s phone number, identification numbers, postal code, etc. are being collected. A nominal variable is one of the 2 types of categorical variables and is the simplest among all the measurement variables. Some examples of nominal variables include gender, name, phone, etc.
* An ordinal variable is a type of measurement variable that takes values with an order or rank. It is the 2nd level of measurement and is an extension of the nominal variable. They are built upon nominal scales by assigning numbers to objects to reflect a rank or ordering on an attribute. Also, there is no standard ordering in the ordinal variable scale. In another sense, we could say the difference in the rank of an ordinal variable is not equal. It is mostly classified as one of the 2 types of categorical variables, while in some cases it is said to be a midpoint between categorical and numerical variables.

1. Terminology

* **Algorithm**: Specific procedure used to implement a particular method or technique on a set of data (e.g. in particular data mining technique). For example, classification three discriminant analysis.
* **Attribute:** Property of an observation.
* **Case**: Observation, Instance.
* **Confidence**: A performance measure in association rules of the type “if A and B are purchased then C is purchased.
* **Dependent variable:** A variable which it value is determined by the value of another variable.
* **Estimation:** Statistical term for analytical term prediction
* **Feature**: A predictor variable
* **Hold out** **data**: A sample of data not used in fitting a model, but instead used to access the performance of the model.
* **Input variable:** A table, observation, or variable used as an input to the predictive model.
* **Model:** An algorithm as applied to a dataset
* **Observation**: The unit of analysis on which the measurements are taken.
* **Outcome variable:** A predicted variable in a supervised technique.
* **Output variable:** Same as outcome variable
* **P(A|B):** The conditional probability of event A occurring given that event B has occurred.
* **Profile:** A set of measurement of an observation
* **Prediction:** The prediction of a numerical value of a continues output variable.
* **Predictor:** A variable usually denoted by X used as an input into a predictive model.
* **Record:** An observation.
* **Score:** A predicted value or class.
* **Supervised learning:** The process of providing an algorithm with records in which, an output variable of interest is known.
* **Target:** Predicted variable.
* **Test data:** The portion of data used only at the end of the model building and selection process to assess how well the final model performs on new data.
* **Training data:** The portion of data used to fit into a model
* **unsupervised Learning**: An analyses in which one attempt to learn patterns in the data other than predicting an output value of interest.
* **Validation data:** The portion of the data used to assess how well the model fits, to adjusts, and to select the best model
* **Variable:** Any measurement on the record relationship between data facts
* Chapter 2 – What is Data Mining
* Relationship between data facts
  + More information lead to more accurate prediction (?)
  + More attributes (observation types) more accurate prediction (?)
  + Relationships among attributes would make a better (accurate) prediction.
  + Target attribute and predictor attributes.
  + Relationships between predictor attributes and target attribute
  + Predictor attributes values cause the target attribute value

Target Attribute Value

. . .

Figure 1: Target variable value as a function of predictors

* Predictor and target attributes relationship.

Actually, the target attribute values are a function of the predictor attributes’ values. So, we can write:

) see figure 1

* Predictive analytics techniques (Supervised Techniques)
  + Classification
  + Regression
* Descriptive analytics and Association Rule (Unsupervised Techniques)

## Data Mining Process

1. Business understanding

In the business understanding phase:

* First, it is required to understand business objectives clearly and find out what are the business’s needs.
* Next, assess the current situation by finding the resources, assumptions, constraints and other important factors which should be considered.
* Then, from the business objectives and current situations, create data mining goals to achieve the business objectives within the current situation.
* Finally, a good data mining plan has to be established to achieve both business and data mining goals. The plan should be as detailed as possible.

1. Data understanding

* The data understanding phase starts with initial data collection, which is collected from available data sources, to help get familiar with the data. Some important activities must be performed including data load and data integration in order to make the data collection successfully.
* Next, the “gross” or “surface” properties of acquired data need to be examined carefully and reported.
* Then, the data needs to be explored by tackling the data mining questions, which can be addressed using querying, reporting, and visualization.
* Finally, the data quality must be examined by answering some important questions such as “Is the acquired data complete?”, “Is there any missing values in the acquired data?”

1. Data preparation

The data preparation typically consumes about 90% of the time of the project. The outcome of the data preparation phase is the final data set. Once available data sources are identified, they need to be selected, cleaned, constructed and formatted into the desired form. The data exploration task at a greater depth may be carried during this phase to notice the patterns based on business understanding.

* 1. Data Exploration – Visualization
  2. Preparation
     1. Dimension Reduction
     2. Records with missing value
     3. Normalizations

1. Modeling

* First, modeling techniques have to be selected to be used for the prepared data set.
* Next, the test scenario must be generated to validate the quality and validity of the model.
* Then, one or more models are created on the prepared data set.
* Finally, models need to be assessed carefully involving stakeholders to make sure that created models are met business initiatives.

1. Evaluation

In the evaluation phase, the model results must be evaluated in the context of business objectives in the first phase. In this phase, new business requirements may be raised due to the new patterns that have been discovered in the model results or from other factors. Gaining business understanding is an iterative process in data mining. The go or no-go decision must be made in this step to move to the deployment phase.

1. Deployment

The knowledge or information, which is gained through data mining process, needs to be presented in such a way that stakeholders can use it when they want it. Based on the business requirements, the deployment phase could be as simple as creating a report or as complex as a repeatable data mining process across the organization. In the deployment phase, the plans for deployment, maintenance, and monitoring have to be created for implementation and also future supports. From the project point of view, the final report of the project needs to summary the project experiences and review the project to see what need to improved created learned lessons.  
These 6 steps describe the Cross-industry standard process for data mining, known as Cross-Industry Standard Process for Data Mining (CRISP-DM). It is an open standard process model that describes common approaches used by data mining experts. It is the most widely-used analytics model.

The figure 2 shows CRISP-DM which, shows the data mining process steps and identifies the relationships among steps

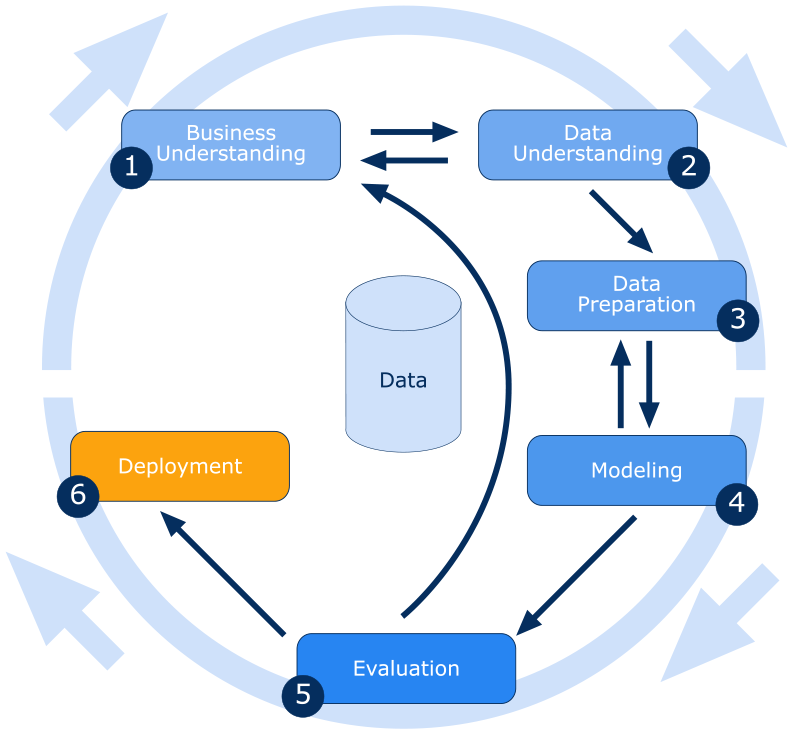


Figure 2: CRISP-DM

## Data Partitioning

Supervised machine learning methods typically require splitting data into multiple chunks for training, validating, and finally testing classifiers. For finding the best parameters of a classifier, training and validation are usually carried out with cross-validation. This is followed by application of the classifier with optimized parameters to a separate test set for estimating the classifier’s generalization performance. With limited data, this separation of test data creates a difficult trade-off between having more statistical power in estimating generalization performance versus choosing better parameters and fitting a better model.

**R Packages**

**Method 1:**

* Caret: is used for partitional data. Since this is already installed with R you need just to load it with library() function.
* The R code for partitioning a dataset into training and testing is: library(caret)

The data name is *vac* and the target attribute is *Result.* You can use this method iff you have a target variable.

set.seed(2015)

#partitioning the data in vac dataset

sample<-createDataPartition(vac$Result, p = 0.7, list = FALSE)

train.a<-vac[sample,]

#validation data

rest<-vac[-sample,]

Since you use a random number generator to generate a record number and select that record, then you need a seed to start the random generator number. In that example the code I chose 2015. The training size is set to 70% (p = 0.7), The “sample” variable holds the record numbers for training dataset. The column MEDV column is specified so, a representative of the target variable be in the training set.

In the consequent codes we move appropriate records (specified in sample list) into train and not specified in sample list ([-sample,])

**Method 2:**

* Here there is no specific package is required since we use stat function *sample* . The data name book and we randomly get the indexes of 60% of data. The we use the indexed to get the training and validation data

#partitioning the data

#training data in the book dataset

set.seed(0)

train.index<-sample(rownames(book), dim(book)[1]\*0.6)

train.df<-book[train.index, ]

#validation data

valid.index <- setdiff(rownames(book), train.index)

valid.df<-book[valid.index, ]

* Chapter 4 –Dimension Reduction

Dimension reduction is the mapping of data to a lower dimensional space such that uninformative variance in the data is discarded, or such that a subspace in which the data lives is detected. Dimension reduction has a long history as a method for data visualization, and for extracting key low dimensional features (for example, the 2-dimensional orientation of an object, from its high dimensional image representation). In some cases, the desired low dimensional features depend on the task at hand. Apart from teaching us about the data, dimension reduction can lead us to better models for inference.

## Data Summaries

Summary statistics summarize and provide information about your [sample](https://www.statisticshowto.datasciencecentral.com/sample/)data. It tells you something about the values in your data set. This includes where the [average](https://www.statisticshowto.datasciencecentral.com/arithmetic-mean/)lies and whether your data is [skewed](https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/skewed-distribution/). Summary statistics fall into three main categories:

* Measures of location (also called [central tendency](https://www.statisticshowto.datasciencecentral.com/central-tendency-2/)).
* [Measures of spread](https://www.statisticshowto.datasciencecentral.com/measures-of-spread/).
* Graphs/charts.

**Summary Statistics: Measures of location**

Measures of location tell you where your data is centered at, or where a trend lies

* Mean (also called the arithmetic mean or average).
* Geometric mean (used for interest rates and other types of growth).
* Trimmed Mean (the mean with outliers excluded).
* Median (the middle of a data set).

**Summary Statistics: Measures of Spread**

Measures of spread tell you (perhaps not surprisingly!) how spread out or varied your data set is. This can be important information. For example, test scores that are in the 60-90 range might be expected while scores in the 20-70 range might indicate a problem. Range isn’t the only measure of spread though. Click on one of the names below for a full definition of that particular measure of spread.

* Range (how spread out your data is).
* Interquartile range (where the “[middle fifty](https://www.statisticshowto.datasciencecentral.com/middle-fifty/)” percent of your data is).
* Quartiles (boundaries for the lowest, middle and upper quarters of data.
* Skewed (does your data have mainly low, or mainly high values?).
* Kurtosis (a measure of how much data is in the tails).

**Summary Statistics: Graphs and Charts**

There are literally dozens of ways to display summary data using graphs or charts. Some of the most common ones are listed below. Click on any name for a definition of that particular chart type.

* Histogram.
* Frequency Distribution Table.
* Box plot.
* Bar chart.
* Scatter plot.
* Pie chart

## Data Probing

* Data Summary (revisited)
  + max(), min(), mean(),and summary R functions.
* Measures of Central Tendency(revisited)
  + Outliers
  + Average
  + Median
  + Mode
* Histogram & Measures of Dispersion
  + Range = max - min
  + Inter-Quartile Range (IQR)

The **interquartile range** (IQR) is a measure of variability, based on dividing a data set into quartiles.

Quartiles divide a rank-ordered data set into four equal parts. The values that divide each part are called the first, second, and third quartiles; and they are denoted by Q1, Q2, and Q3, respectively.

* Q1 is the "middle" value in the *first* half of the rank-ordered data set.
* Q2 is the median value in the set.
* Q3 is the "middle" value in the *second* half of the rank-ordered data set.

The interquartile range is equal to Q3 minus Q1.

For example, consider the following numbers: 1, 3, 4, 5, 5, 6, 7, 11. Q1 is the middle value in the first half of the data set. Since there are an even number of data points in the first half of the data set, the middle value is the average of the two middle values; that is, Q1 = (3 + 4)/2 or Q1 = 3.5. Q3 is the middle value in the second half of the data set. Again, since the second half of the data set has an even number of observations, the middle value is the average of the two middle values; that is, Q3 = (6 + 7)/2 or Q3 = 6.5. The interquartile range is Q3 minus Q1, so IQR = 6.5 - 3.5 = 3.

In some texts, the interquartile range is defined differently. It is defined as the difference between the largest and smallest values in the middle 50% of a set of data.

To compute an interquartile range using this definition, first remove observations from the lower quartile. Then, remove observations from the upper quartile. Then, from the remaining observations, compute the difference between the largest and smallest values.

For example, consider the following numbers: 1, 3, 4, 5, 5, 6, 7, 11. After we remove observations from the lower and upper quartiles, we are left with: 4, 5, 5, 6. The interquartile range (IQR) would be 6 - 4 = 2

The figure. 3 shows the 1st, 2nd, 3rd, and 4th, Quartiles (σ is standard deviation)

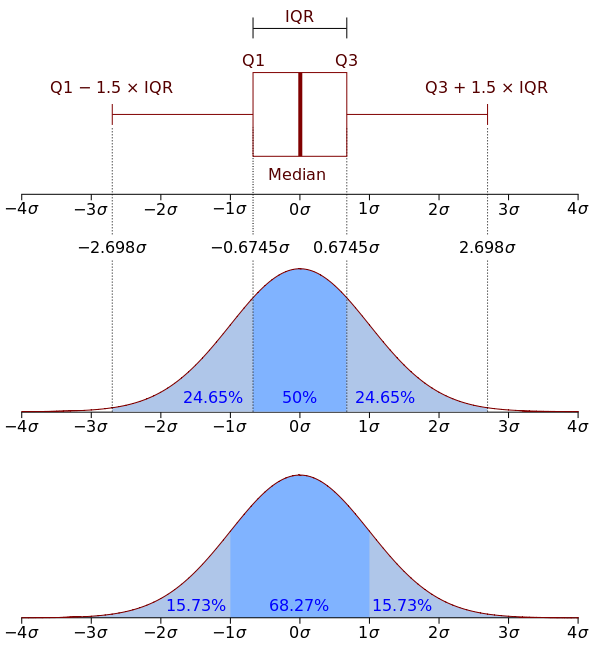


Figure 3: Quartile

* Variance and Standard Deviation
  + What is the variance and how it is calculated?
  + What is standard deviation and how it is calculated?
  + Coefficient of Variance

* Data Characteristics
  + Data Skewness and relationship between mean and median

In probability theory and statistics, skewness is a measure of the asymmetry of the probability distribution of a real-valued [random variable](https://en.wikipedia.org/wiki/Random_variable) about its mean. The skewness value can be positive or negative, or undefined.

Consider the two distributions in the figure 2. Within each graph, the values on the right side of the distribution taper differently from the values on the left side. These tapering sides are called tails, and they provide a visual means to determine which of the two kinds of skewness a distribution has:

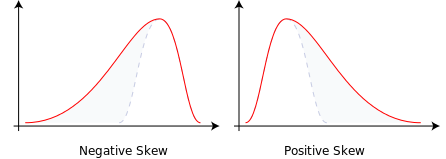
1. **negative skew**: The left tail is longer; the mass of the distribution is concentrated on the right of the figure. The distribution is said to be left-skewed, left-tailed, or skewed to the left, despite the fact that the curve itself appears to be skewed or leaning to the right; left instead refers to the left tail being drawn out and, often, the mean being skewed to the left of a typical center of the data. A left-skewed distribution usually appears as a right-leaning curve.
2. **positive skew**: The right tail is longer; the mass of the distribution is concentrated on the left of the figure. The distribution is said to be right-skewed, right-tailed, or skewed to the right, despite the fact that the curve itself appears to be skewed or leaning to the left; right instead refers to the right tail being drawn out and, often, the mean being skewed to the right of a typical center of the data. A right-skewed distribution usually appears as a left-leaning curve

Figure 2: Data Skewness

* + Data Plotting – First step in data mining
  + Measure of relationship among attributes (between target and predictor)
    - Covariance
    - Coefficient of Correlation
* R Dealing with empty cells in a table
* Chapter 4 – Using Correlation and Covariance to Reduce Dataset Dimensions

**Correlation Analysis**

Correlation is a statistical method used to evaluate the strength of relationship between two quantitative variables or two dimensions. A high correlation means that two or more variables have a strong relationship with each other, while a weak correlation means that the variables are hardly related.

Dimension reduction is an important topic in data mining and machine learning. By analyzing correlation matrix we can identified those pairs of variables which share same characteristics such as variance. In this case we may use one of the dimensions instead of both for predictive process.

Another method is using covariance between pair of dimension and use a new dimension with characteristics of both variables. This method is called Principal Component Analysis

* Principal Component Analysis

**Covariance**

Before we get started, we shall take a quick look at the difference between covariance and variance. Variance measures the variation of a single random variable (like the height of a person in a population), whereas covariance is a measure of how much two random variables vary together (like the height of a person and the weight of a person in a population). The formula for variance is given by the following equation where n is the number of observations

The covariance σ(x,y) of two random variables x and y is given by

with n samples. The variance of dimension x can be also expressed as the covariance with itself by

**Covariance Matrix**

With the covariance we can calculate entries of the covariance matrix of two dimensions, which is a square matrix given by *C:*

C =

Which is

C =

And cov =

Then we can write

## Principal Component Analysis

Principal Component Analysis (PCA) is a simple yet popular and useful linear transformation technique that is used in numerous applications, such as stock market predictions, the analysis of gene expression data, and many more areas where the number of dimensions are large or very large. In this course we just learn how covariance of two dimensions could be used to compute two new dimensions *Z1* and *Z2*, which are called first and second PCAs. Then we replace the two original dimensions with only the *Z1* which is the First PC.

Let’s repeat this!

* Covariance of two dimensions measures the variability of the two dimensions when are considered together. *Z1* is linear transformation of the maximum variability of the two dimensions when considered together.
* *Z2* is orthogonal to *Z1* and is the second largest variability of our two dimensions.
* Information captured in *Z1* and *Z2* are uncorrelated to each other.
* So we will pick the *Z1* as the substitution to our original dimensions.

To apply this algorithm to a dataset with large number of variable we must first normalize the dataset.

Check our textbook for R code for normalization of the dataset and creating PC of all dimensions to each other.

* Data components and R
* Useful link for Data Science with R. Check first chapter for all about RStudio

<https://rafalab.github.io/dsbook/>

## Useful R Codes

**Str()**

str() is a compact way to display the structure of an R object.

This allows you to use str as a diagnostic function and an alternative to summary. str will output the information on one line for each basic structure. str is best for displaying contents of lists. The goals is to get an output for any R object.

**class()**

class() is an easy way to display the class of an R object.

**Missing Data**

In R, missing values are represented by the symbol NA (not available). Impossible values (e.g., dividing by zero) are represented by the symbol NaN (not a number). Unlike SAS, R uses the same symbol for character and numeric data.

* is.na() for testing for missing values
* na.omit() for removing missing values

**options()**

Description

Allow the user to set and examine a variety of global “options” which affect the way in which R computes and displays its results.

options(digits = n)

controls the number of digits to print (display) when printing numeric values. It is a suggestion only. Valid values are 1...22 with default 7.

**prcom() and princomp()**

The simplified format of these 2 functions are :

prcomp(x, scale = FALSE)

princomp(x, cor = FALSE, scores = TRUE)

Arguments for prcomp():

* x: a numeric matrix or data frame
* scale: a logical value indicating whether the variables should be scaled to have unit variance before the analysis takes place

Arguments for princomp():

* x: a numeric matrix or data frame
* cor: a logical value. If TRUE, the data will be centered and scaled before the analysis
* scores: a logical value. If TRUE, the coordinates on each principal component are calculated

The elements of the outputs returned by the functions prcomp() and princomp() includes :

|  |  |  |
| --- | --- | --- |
| prcomp() name | princomp() name | Description |
| sdev | sdev | the standard deviations of the principal components |
| rotation | loadings | the matrix of variable loadings (columns are eigenvectors) |
| center | center | the variable means (means that were substracted) |
| scale | scale | the variable standard deviations (the scaling applied to each variable ) |
| x | scores | The coordinates of the individuals (observations) on the principal components. |

**Codes for PCA output visualization**

install visualization package for github1

install.packages("factoextra")

library(factoextra)

fviz\_eig(pcs.cov.2)

fviz\_pca\_ind(pcs.cov.2, col.ind = "cos2", gradient.cols = c("#00AFBB", "#E7BB00", "#FC4E07"), repel = TRUE)

Where in this example pcs.cov.2 is the output of prcomp() function.

And the following is the grapg of the fviz\_eig90 function

