**STAT6620-01- Homework1**

**CHINKI RAI**

**03/29/2017**

1. **Definitions**

**Statistical Learning:**

From Wikipedia- statistical learning theory is a framework for machine learning drawing from the fields of statistics and functional analysis.

By Thomas Minka- An approach to machine intelligence which is based on statistical modeling of data. With a statistical model in hand, one applies probability theory and decision theory to get an algorithm.

From spinger book- Statistical learning refers to a set of tools for modeling and understanding complex datasets.

**Statistical Machine Learning:** Statistics machine learning is statistics with computational sciences.

**Machine Learning:** An algorithm that can learn from data without relying on rules-based programming. Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed.

**Predictive Analytics:** Predictive analytics is the practice of extracting information from existing data sets in order to determine patterns and predict future outcomes and trends. Predictive analytics does not tell you what will happen in the future. Predictive analytics is the branch of the advanced analytics which is used to make predictions about unknown future events. Predictive analytics uses many techniques from data mining, statistics, modeling, machine learning, and artificial intelligence to analyze current data to make predictions about future.

**Artifical Intelligence:** The theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.AI is Making computational models of human behavior.

**Deep Learning:** Deep learning (also known as deep structured learning, hierarchical learning or deep machine learning) is a class of machine learning algorithms. Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.

**Big Data:** Extremely large data sets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions. Big data is a term that describes the large volume of data both structured and unstructured that inundates a business on a day-to-day basis.

1. **Chapter second code**

* subject\_name <- c("John Doe", "Jane Doe", "Steve Graves")

Subject\_name is a vector which store characters.

* temperature <- c(98.1, 98.6, 101.4)

temperature is a double vactor which store store each patient's body temperature.

* flu\_status <- c(FALSE, FALSE, TRUE)

flu\_status is a logical vector store each patient's diagnosis.

* temperature[2]

98.6

extracting second value of vector temperature.

* temperature[2:3]

98.6 101.4

extracting second to third value of vector temperature.

* temperature[-2]

98.1 101.4

excluding second value from vector temperature.

* temperature[c(TRUE, TRUE, FALSE)]

98.1 98.6

To include the first two temperature readings but exclude the third, type.

* gender <- factor(c("MALE", "FEMALE", "MALE"))

> gender

[1] MALE FEMALE MALE

Levels: FEMALE MALE

To create a factor from a character vector. factor function.

* blood <- factor(c("O", "AB", "A"),

levels = c("A", "B", "AB", "O"))

blood

[1] O AB A

Levels: A B AB O

Creating a factor with additional level.

* symptoms <- factor(c("SEVERE", "MILD", "MODERATE"), levels = c("MILD", "MODERATE", "SEVERE"), ordered = TRUE)

symptoms

[1] SEVERE MILD MODERATE

Levels: MILD < MODERATE < SEVERE

Order represent sequential order of factor from mild to severe.

* symptoms > "MODERATE"

[1] TRUE FALSE FALSE

we can test whether each patient's symptoms are greater than mo derate.

* subject\_name[1]

"John Doe"

Representing first value of list subject\_name.

* temperature[1]

[1] 98.1

Representing first value of list temperature.

* flu\_status[1]

[1] FALSE

* gender[1]

[1] MALE

Levels: FEMALE MALE

* blood[1]

[1] O

Levels: A B AB O

* symptoms[1]

[1] SEVERE

Levels: MILD < MODERATE < SEVERE

* subject1 <- list(fullname = subject\_name[1],temperature = tempera ture[1],flu\_status = flu\_status[1],gender = gender[1],blood = blood[1],symptoms = symptoms[1])

> subject1

$fullname

[1] "John Doe"

$temperature

[1] 98.1

$flu\_status

[1] FALSE

$gender

[1] MALE

Levels: FEMALE MALE

$blood

[1] O

Levels: A B AB O

$symptoms

[1] SEVERE

Levels: MILD < MODERATE < SEVERE

To create a list with named components for all of the first

patient 's data.

* subject1[2]

$temperature

[1] 98.1

To access the temperature value from list subject.

* > subject1[[2]]

[1] 98.1

code returned a list with a single temperature component.

* subject1$temperature

[1] 98.1

access list components directly, by appending a $ and the value's name to the name of the list component.

* subject1[c("temperature", "flu\_status")]

$temperature

[1] 98.1

$flu\_status

[1] FALSE

returns a subset of the subject1 list, which contains only the temperature and flu\_status components.

**DATA FRAMES**

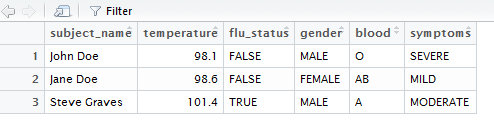
In R terms, a data frame can be understood as a list of vectors or factors, each having exactly the same number of values. it combines aspects of both vectors and lists.

* pt\_data <- data.frame(subject\_name, temperature, flu\_status, gender, blood, symptoms, stringsAsFactors = FALSE)

stringsAsFactors = FALSE means character vectors should not converted to

factors. Here subject\_nam is not categorical data.

Creating a data frame with list of vectors.



* pt\_data$subject\_name

[1] "John Doe" "Jane Doe" "Steve Graves"

to obtain the subject\_name vector, type.

* pt\_data[c("temperature", "flu\_status")]

temperature flu\_status

1 98.1 FALSE

2 98.6 FALSE

3 101.4 TRUE

a vector of names can be used to extract several columns from a

data frame.

* pt\_data[2:3]

temperature flu\_status

1 98.1 FALSE

2 98.6 FALSE

3 101.4 TRUE

Second and third vector is extract from data frame.

* pt\_data[1, 2]

[1] 98.1

extract the value in the first row and second column of the patient data frame.

* pt\_data[c(1, 3), c(2, 4)]

temperature gender

1 98.1 MALE

3 101.4 MALE

will pull data from the first and third rows and the second and fourth columns.

* pt\_data[, 1]

[1] "John Doe" "Jane Doe" "Steve Graves"

To extract all the rows of the first column.

|  |
| --- |
| * pt\_data[1, ]   subject\_name temperature flu\_status gender blood symptoms  1 John Doe 98.1 FALSE MALE O SEVERE |

|  |
| --- |
| * pt\_data[ , ]   subject\_name temperature flu\_status gender blood symptoms  1 John Doe 98.1 FALSE MALE O SEVERE  2 Jane Doe 98.6 FALSE FEMALE AB MILD  3 Steve Graves 101.4 TRUE MALE A MODERATE  To extract everything, use the following command. |
|  |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | * pt\_data[c(1, 3), c("temperature", "gender")]   temperature gender  1 98.1 MALE  3 101.4 MALE  We can use pt\_data[-2, c(-1, -3, -5, -6) for same output as above.  MATRIXES AND ARRAYS:  A matrix is a data structure that represents a two-dimensional table with rows and columns of data. Like vector it contains only one type of data.   |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | * m <- matrix(c(1, 2, 3, 4), nrow = 2)   m  [,1] [,2]  [1,] 1 3  [2,] 2 4  to create a 2 x 2 matrix storing the numbers one through four, we can use  the nrow parameter to request the data to be divided into two rows.  m <- matrix(c(1, 2, 3, 4), ncol = 2) will get same matrix.   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | * m <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2 )   m  [,1] [,2] [,3]  [1,] 1 3 5  [2,] 2 4 6  With six values, requesting two rows creates a matrix with three columns.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | * m <- matrix(c(1, 2, 3, 4, 5, 6), ncol = 2)   m  [,1] [,2]  [1,] 1 4  [2,] 2 5  [3,] 3 6  Requesting two columns creates a matrix with three rows.   * m[1, ]   [1] 1 4  To call 1st row of the matrix m.   * m[, 1]   [1] 1 2 3  To call 1st column of the matrix m.  **MANAGING DATA WITH R**.     * Save the objects   x=1  y=2  z=2  save(x, y, z, file = "mydata.RData")  load("mydata.RData")  We have three objects x, y ,z that we would like to save in a  permanent file. Regardless of whether they are vectors, factors,  lists, or data frames, we could save them to a file named mydata.RData  R data files have an .RData extension. The load() command can recreate  any data structures that have been saved to an .RData file.  the save.image() command will write your entire session to a file simply  called .RData.   * ls()   [1] "a" "b" "blood" "c"  [5] "flu\_status" "gender" "hello" "m"  [9] "mushroom\_1R" "mushroom\_c5rules" "mushroom\_JRip" "mushrooms"  [13] "pt\_data" "subject\_name" "subject1" "symptoms"  [17] "temperature" "wd" "word" "x"  [21] "y" "z"  The ls() listing function returns a vector of all the data structures currently in the memory.   * rm(m, subject1)   The rm() remove function can be used to remove m and subject1.   * rm(list=ls())   This works with the ls() function to clear the entire R session.  **IMPORTING AND SAVING DATA FROM CSV files**   * To import a csv file   Usedcars<-read.csv("usedcars.csv",stringsAsFactors = FALSE)   * To save a data frame to csv.   write.csv(usedcars, file = " usedcars.csv ", row.names = FALSE)    **Exploring the structure of Data**   * str(usedcars)   'data.frame': 150 obs. of 6 variables:  $ year : int 2011 2011 2011 2011 2012 2010 2011 2010 2011 2010 ...  $ model : chr "SEL" "SEL" "SEL" "SEL" ...  $ price : int 21992 20995 19995 17809 17500 17495 17000 16995 16995 16995 ...  $ mileage : int 7413 10926 7351 11613 8367 25125 27393 21026 32655 3616 ...  $ color : chr "Yellow" "Gray" "Silver" "Gray" ...  The str() function provides a method to display the structure of R data structures such as data frames, vectors, or lists. Number of variables and  Type of variable.   * summary(usedcars$year)   Min. 1st Qu. Median Mean 3rd Qu. Max.  2000 2008 2009 2009 2010 2012  Summary is function use to get mean, median, quartiles, min and max  Value.   * summary(usedcars[c("price", "mileage")])   price mileage  Min. : 3800 Min. : 4867  1st Qu.:10995 1st Qu.: 27200  Median :13592 Median : 36385  Mean :12962 Mean : 44261  3rd Qu.:14904 3rd Qu.: 55125  Max. :21992 Max. :151479  This will give summary of price and mileage column.   * mean(usedcars$price)   [1] 12961.93   * range(usedcars$price)   [1] 3800 21992   * diff(range(usedcars$price))   [1] 18192   * IQR(usedcars$price)   [1] 3909.5   * quantile(usedcars$price)   0% 25% 50% 75% 100%  3800.0 10995.0 13591.5 14904.5 21992.0   * quantile(usedcars$price, probs = c(0.01, 0.99))   1% 99%  5428.69 20505.00   * quantile(usedcars$price, seq(from = 0, to = 1, by = 0.20))   0% 20% 40% 60% 80% 100%  3800.0 10759.4 12993.8 13992.0 14999.0 21992.0  **VISUALIZING VARIABLES**   * Boxplot   boxplot(usedcars$price, main="Boxplot of Used Car Prices",  ylab="Price ($) ",col="Blue")  boxplot(usedcars$mileage, main="Boxplot of Used Car Mileage",  ylab="Odometer (mi.)",col="red")     * Histogram   hist(usedcars$price, main = "Histogram of Used Car Prices",  xlab = "Price ($)",col="red")  hist(usedcars$mileage, main = "Histogram of Used Car Mileage",  xlab = "Odometer (mi.)",col="blue")       * var(usedcars$price)   [1] 9749892     |  | | --- | | sd(usedcars$price)  [1] 3122.482   * table(usedcars$year)   2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012  3 1 1 1 3 2 6 11 14 42 49 16 1   * table(usedcars$model)   SE SEL SES  78 23 49   * model\_table <- table(usedcars$model)   prop.table(model\_table)  SE SEL SES  0.5200000 0.1533333 0.3266667  The results of prop.table() can be combined with other R functions  to transform the output.   * Scatter plot   plot(x = usedcars$mileage, y = usedcars$price, main = "Scatterplot  of Price vs. Mileage", xlab = "Used Car Odometer (mi.)",  ylab = "Used Car Price ($)")       * usedcars$conservative =usedcars$color %in% c("Black", "Gray", "Silver",   "White")  A new command here: the %in% operator returns TRUE or FALSE for  each value in the vector on the left-hand side of the operator  depending on whether the value is found in the vector on the  right-hand side.   * table(usedcars$conservative)   FALSE TRUE  51 99   * library(gmodels)   CrossTable(x = usedcars$model, y = usedcars$conservative)  Cell Contents  |-------------------------|  | N |  | Chi-square contribution |  | N / Row Total |  | N / Col Total |  | N / Table Total |  |-------------------------|  Total Observations in Table: 150  | usedcars$conservative  usedcars$model | FALSE | TRUE | Row Total |  ---------------|-----------|-----------|-----------|  SE | 27 | 51 | 78 |  | 0.009 | 0.004 | |  | 0.346 | 0.654 | 0.520 |  | 0.529 | 0.515 | |  | 0.180 | 0.340 | |  ---------------|-----------|-----------|-----------|  SEL | 7 | 16 | 23 |  | 0.086 | 0.044 | |  | 0.304 | 0.696 | 0.153 |  | 0.137 | 0.162 | |  | 0.047 | 0.107 | |  ---------------|-----------|-----------|-----------|  SES | 17 | 32 | 49 |  | 0.007 | 0.004 | |  | 0.347 | 0.653 | 0.327 |  | 0.333 | 0.323 | |  | 0.113 | 0.213 | |  ---------------|-----------|-----------|-----------|  Column Total | 51 | 99 | 150 |  | 0.340 | 0.660 | |  ---------------|-----------|-----------|-----------|  3.**Statistical Learning**  **Statistical Learning:**  Statistical learning refers to a set of tools for modeling and understanding complex  datasets.It is a recently developed area in statistics and blends with parallel developments  in computer science and, in particular, machine learning. Statistical learning refers to a  vast set of tools for understanding data. These tools can be classiﬁed as supervised or  unsupervised. Broadly speaking, supervised statistical learning involves building a statistical  model for predicting, or estimating, an output based on one or more inputs. | |  | | |  | | --- | |  | | | |  | | |  | | --- | |  | | | |  | | |  | | --- | |  | | | |  | | |  | | --- | |  | | | |  | | |  | | --- | |  | | | |