Practical Machine Learning with R

Lecture 1

Objectives:

- Introduction to R programming Language for Machine Learning.
- Installation of R and RStudio. R Studio basics.
- Installation of packages from CRAN, Bioconductor and Github.
- R Introduction
- Fundamental concepts of Machine Learning.

Introduction To R

Origins:

- The R programming language was created by Ross Ihaka and Robert Gentleman at University of Auckland, New Zealand in 1996.R programming is a one the most powerful language used in the scientific world as well as domain areas related to research and exploratory analysis.

- R programming language was remodeled utilizing the programming paradigms and principles of S, an object oriented language developed by John Chambers at the Bell Labs in 1970. Historically in 1970s S, was implemented for research and scientific endeavors but was not open source application.

Characteristics:

- **Open Source Framework**: R works under the GNU (General Public License) license which is not restrictive in its scope allowing any individual to download and implement it. It has its own community based features and forums.

- **Flexible**: Unlike many proprietary scientific and software systems R is not composed as a rigid architecture that mandates the analysis to be conducted only via a given schema. The analysis framework is not didactic in its implementation wherein the only way to conduct research based tests is through a menu driven graphical interface. Packages can be created by developers for different tasks as well as newer functions can be customized to achieve the objective of the research.

- **Extensible**: R is tailored to allow research not only in the domain areas of statistics but also provides packages and utilities for Bio informatics, Language processing, humanities, econometric, geo spatial arena to name a few. R implements the Object Oriented approach following in the footsteps of its parent programming language S.

- **Comprehensive:** R incorporates the full spectrum of computational and visualization frame works that allow for holistic and novel research and analysis. It provides plethora of diverse tools and techniques to implement the full data science or machine learning project lifecycle from inception to completion. This exhaustive lifecycle entails importing

data, cleaning data, normalizing data , algorithmical analysis of data, validation as well as efficacy determination.

- **Cross Platform Compatible:** The R programming language can be implemented seamlessly across platforms ranging from Windows OS, Mac OS,UNIX, Solaris to name a few. R programming has been used to created many industry based application and companies like Google, Oracle, SAP are leveraging the power of this language. It can also be implemented in conjunction with other languages like Python.

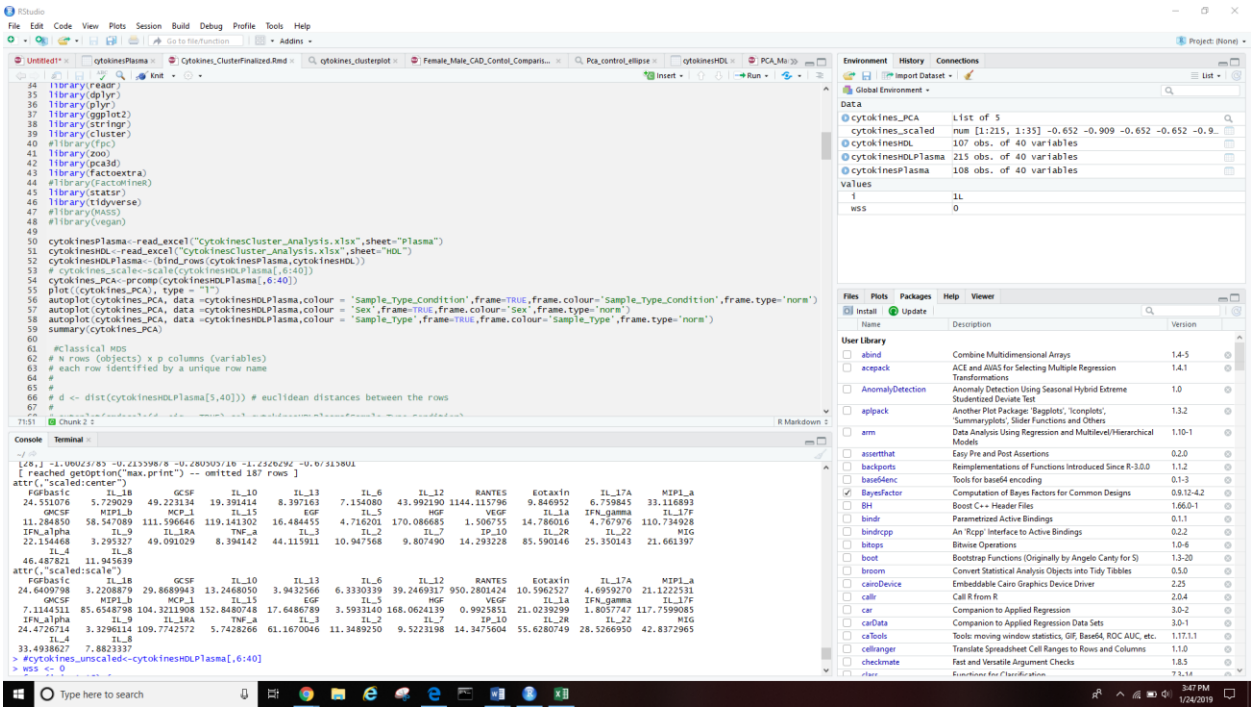
Installation of R and RStudio

Steps:

1) Browse to the **CRAN** website using the link: https://cran.r-project.org/
2) Choose and download the precompiled binaries as per the operating system of the user. The available operating systems are MacOS, Windows as well as Linux. Newer versions are published periodically therefore download the latest ones.
3) Follow the menu driven instructions to complete the installation of the R Software.
4) To download and install RStudio IDE(Integrated Development Environment) browse to the website: https://www.rstudio.com/products/rstudio/download/
5) Choose the Free version as per your Operating System and follow the installer direction to install RStudio

RStudio

RStudio console is composed of four windows : ***Script editor, Workspace environment, Command line Console and Misc displays.***



The top left window is the Script editor that contains the script of the code. It also incorporates features that help save the files using different formats, syntax highlighting , commenting as well as other enhanced features.

The top right window is the Workspace environment window that displays and saves detailed information related to the contextual data frames, vectors as well as the user defined objects.

Console window is at the lower left window. It serves as the space where the output of the code is displayed. A developer can also run a line code scripts.

The Misc displays is the lower right window The tabs provides with the information related to the files in the working directory, plots generated by the code, packages downloaded and installed as well as the help option .For a function help we can use the command >help(function).

R Notebook framework is very useful since it has several chunks that can be combined to create a journal, book etc where diverse elements like images, code outputs, mark up as well as latex for formula can be incorporated.

Packages

Packages are bundles which encapsulate code, documentation, tests as well as data. Packages are created by R developers and uploaded on the CRAN, Biconductor as well as Github. These packages can be downloaded by any individual from these website . As of now there are more than 2000 packages.

Installation of packages from CRAN, Bioconductor and Github.

CRAN:

- install.packages ("packagename")

BICONDUCTOR:

The command to get to the Bi conductor website is as follows:

- source("http://bioconductor.org/biocLite.R")

The command to install standard packages:

- biocLite()

The command to install specific packages:

- biocLite("packagename")

GitHub:

To install Github devtools package is used. The command is as follows:

- install.packages("devtools")
-

The command to install a specific package is as follows:

- devtools::install_github("username/packagename")

Package Loading

The command to load a package is as follows:

- library(packagename)

To obtain help for a package

- help(package=”packagename””)

R INTRODUCTION

Every programming language has different Syntax (command grammar) and Semantics (Interpretation of the command).

- R is an object-oriented language. Everything in R are objects.
- The main data structure in R is a vector. a<-4 is a 1*1 vector.
  v←c(2,4,5,6) is a 1*4 vector
  This is assigning the values to the vector v

  ← is an assigning operator
- All the commands are functions for example help() is a function
- Some commands can have multiple outputs depending on the input arguments.

Main Data Structures

1) Vectors

Stores numerous elements. They can comprise of numbers, text , logical variables etc.

Vector data structure cannot simultaneously incorporate more than one data type . For instance ,a vector cannot contain elements that are integers as well as text , both at the same time.

- v←c(“Hello”,”Hi”,”Welcome”)
- v

  “Hello”,”Hi”,”Welcome”

2) List

List stores elements that might be either of the same type or of different types.

Record<-list(name=”mary”,gpa=4,major=”History”)

To access an individuals record we have to provide the index.

3) Matrices and Arrays

Matrix is a two-dimensional framework with specific elements at the given indices. These elements in a matrix can be of one type only. Usually matrices are used for numerical computation therefore the stored data type is typically numerical.

Student_grades←matrix(100,200,150,180,ncol=2)

Student_grades

```
     [,1] [,2]
[1,]  100  150
[2,]  200  180
```

4) Arrays

Are structures like matrices where the structure can be multidimensional.

5) Data Frames

Data Frames are the most paramount data structure within the R programming environment as it combines the functionality of lists and vectors. A data frame is analogous to a spreadsheet where it can consist of elements that are of different data types.

- Student_frame←data.frame(student_name, student_grade,studentgpa,stringsAsFactors=FALSE)

If we do not set the stringsAsFactors=FALSE then R converts every character vector to a factor.

- Student_frame

Student Name	Student grade	studentgpa

Data frames are very advantageous since they are displayed as a 2 dimensional matrix which can showcase disparate data types at the same time.

The columns of the data frames are defined as attributes or features and the rows are defined as examples.

Factors

Data can be either numerical or categorical. Categorical data can be further differentiated into nominal and ordinal type. Features of a data set are Nominal if there is no natural ordering wheras features of a data set are ordinal if there is a natural ordering in the dataset.

If data inherently is nominal or ordinal data then R uses a special data type called factors that saves the data with categories. Hypothetically, if we have a dataframe that has a column which defines the gender type Male and Female then R will create two categories. Since there are recurring records pertaining to males or females, the category label for male(male:label=1) and female(female:label=2) is stored once and subsequently this is stored as numbers 1,2,1,2 to identify every example/row. The R framework will automatically map 1 to Male and 2 to Female. This strategy not only saves the memory but also informs the reader that the data type has categories.

Gender<-factor (c("Male"."Female","Male"))

Gender

Male Female Male

Levels: Male Female

Useful Inbuilt Functions in R for exploring data features

- View Data Structures in Memory
  - ls()
- Remove data structure from memory
  - rm(Student_frame)
  - rm(list=ls())
- Reading Data from a file into a data frame
  - Student_record ← read.csv("student.csv", stringsAsFactors = FALSE)
- Visualizing seeing the structure of a data frame
  - str(Student_record)
- Summarizing the data in the data frame
  - summary(Student_record)
  - summary(Student_record$gpa)
  - summary(Student_record[c("gpa","major")])
- mean
  - mean(Student_record$gpa)
- median

- median(Student_record$gpa)

- range
  - range(Student_record$gpa)
- IQR
  - IQR(Student_record$gpa)
- Boxplot
  - boxplot(Student_record$gpa, main="Boxplot of Used Student gpa", ylab = "GPA")
- Histogram
  - hist(Student_record$gpa, main="Boxplot of Used Student gpa", ylab = "GPA")
- Variance, Sd
  - var(Student_record$gpa)
  - sd(Student_record$gpa)
- One way Table
  - table(Student_record$major)
- Proportions
  - major_table ← table(Student_record$major)
  - prop.table(major_table)
- Rounding
  - round(major_table, digits = 1)
- Scatterplot
  - plot(x = Student_record$gpa, y = Student_record$grade,

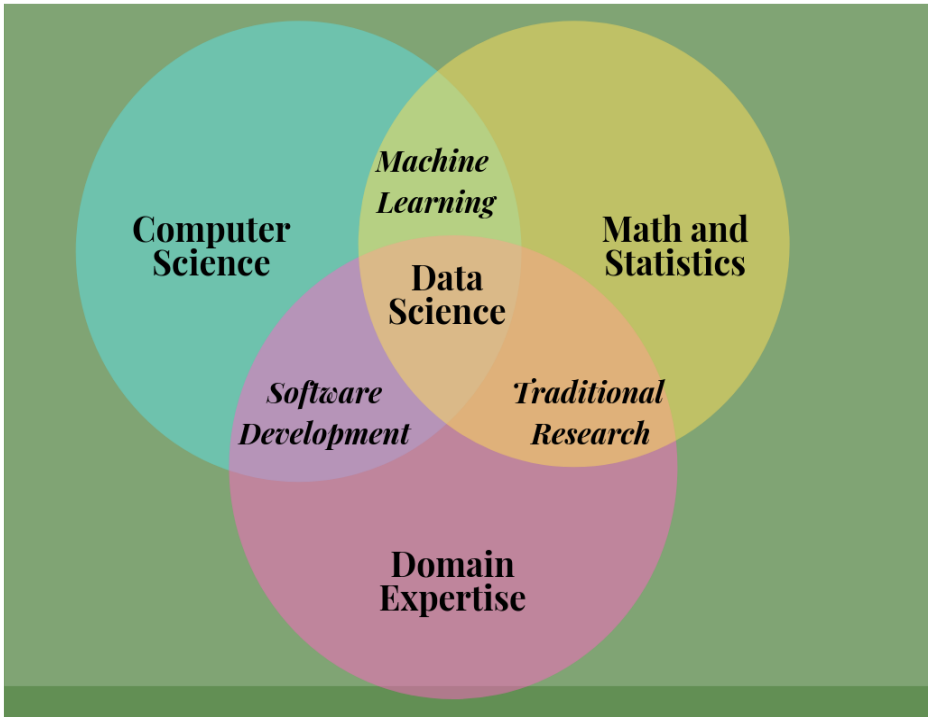
```
main = "Scatterplot of gpa and grade",
xlab = "grade",
ylab = "gpa")
```

- Crosstabulation table
    - CrossTable(x = Student_record$major, y = Student_record $full_time)

Foundation of Data Science --- An Introduction:

- In contemporary times the digital progression has caused a proliferation of data and data is electronically available across a whole spectrum media sources pertaining to diverse domain areas.
- Complex and large volume of data available must be translated into substantive inferences and actionable insights for it to be of any value. Data Science serves the explicit purpose of translating data into knowledge.
- Data Scientist utilizes numerous techniques from the field of Statistics, Mathematics , Computer Science, Machine Learning and Artificial Intelligence to convert information into knowledge.
- Data Science is a cross disciplinary set of skills that are applied to solve complex problems across diverse disciplines.

Visualization of Data Science domain specifications:



Fundamental Knowledge and Taxanomical Frameworks of Machine Learning

Machine learning is a repertoire of diverse algorithms that enable gaining substantive insights from the contextual data being analyzed which in turn can be leveraged for the decision making in manifold domain areas.

Machine learning is prevalently used in the field of Data Analytics which has proliferated the research sectors of society due to the advancement of information and technology which caused the generation of huge recorded data sets with disparate internal structures.

The big data which is characterized by attributes of volume, variety and velocity in conjunction with high computing power of the computer systems and the statistical models facilitated the advent and ubiquitous application of Machine learning algorithms.

Machine learning algorithm teaches the computer how to use the data to solve a problem.

The computational power of computers for implementation of Machine Learning can only be harnessed and utilized for Real life application by human intervention and direction. The process can only be initiated by humans as well as progressive incremental actions are at the discretion of humans. The algorithms might not be able to extrapolate and does not have the intuition of the human brain. In addition to the fact that it is not able to fully work like the brain of a human being , it also is not competent to always create and translate grammatically correct constructs.

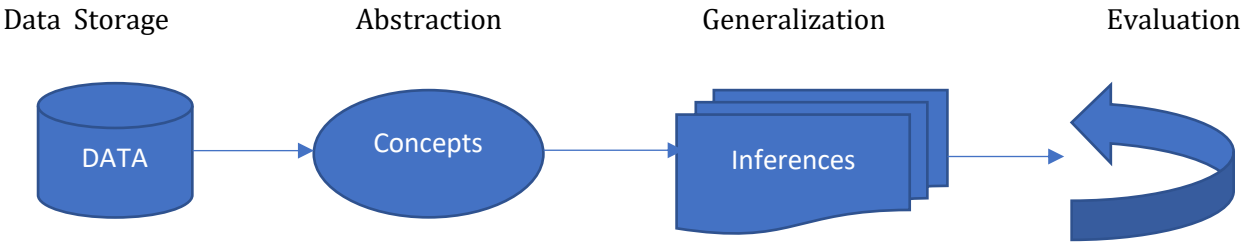
The Machine learning Algorithms is complimentary to the domain knowledge expert therefore ameliorated the decision making in a given scenario. A machine learning algorithm will help a medical doctor visually and numerically perceive the patterns in the biomarkers distributions consequent to which the doctor can make informed decisions related to his/her patient set.

Some important applications of machine learning include fraud detection, disease pathway exploration, weather prediction, auto pilot frameworks, crime pattern assessment etc

Machine Learning Process:

The machine learning process primarily comprises of four germane and interrelated components. These can be elucidated as follows:

1) Data Storage
2) Abstraction
3) Generalizaton
4) Evaluation



Data Storage:

The first component of the learning process is the contextual data in various storage devices and in various formats. Some of this data might be fully structured like a data base file from the Oracle company or it might be a unstructured data from a sequence of click stream.

Abstraction

The second component of the learning process is the conversion of the raw data into explicit knowledge representation and insightful patterns by mapping the data to a specific model . The model categories are as follows:

- Mathematical equations
- Relational diagrams
- Logical if/else
- Grouping into clusters

The conversion process entails fitting the model to the data set which is defined as Training. Subsequent to the training the data represents and summarizes the original data in a more abstract manner. An illustration to clarify this step is the generation of a linear regression equation from a data set composed of two correlated columns.

Generalization

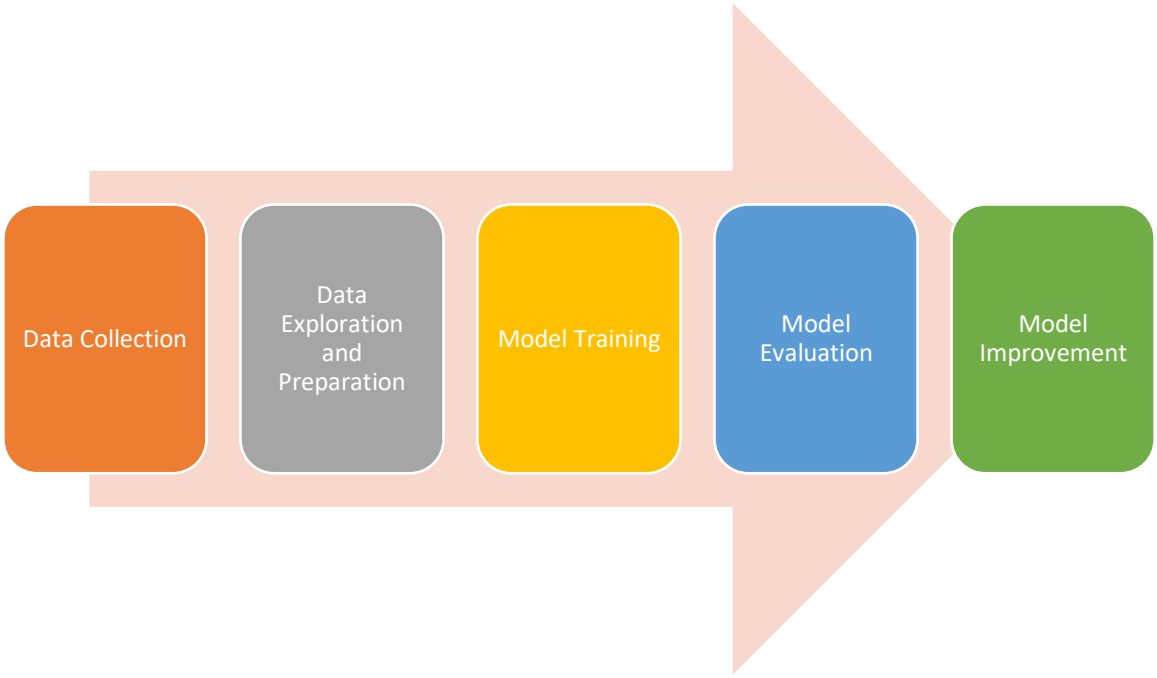
The third component of the learning process is Generalization which entails targeted identification of patterns that align with the objective of the study or are significant in comparison to all possible patterns to effectuate the decision process. After the algorithm completes its run it heuristically (short cut mechanisms) might conjecture related to the most useful inferences. If the heuristics applied are errorneous then it is defined as bias. For instance if a human is identified as standing on two feet then a toddler will not be identified as a human. Bias needs to be minimized in order to be able to make substantive and credible decisions.

Evaluation

Is the fourth component in the learning process which ascertains the success of the machine learning algorithm to inform decision making in the face of the varying levels of bias observed inherently in all of them. Subsequent to the training of the data ,t he model created by the generalization process is tested on a new data set defined as the testing data. This will determine how well the characterization of the training data generalizes to the testing data. The generalization mapping can never be a perfect mapping but data scientist endeavor to obtain the best fitted model.

The generalization is not perfect due to the explained and unexplainable variations in the data. The reasons for these variations might include errorneous measurements, missing data or complexity of the phenomenon. It is important to identify the underlying pattern without taking cognizance of the underlying noise because if we model the noise defined as overfitting , the generalization to the test data will inaccurate. If a model works well during training but categorizes imperfectly during testing is due to the issue of overfitting.

Lifecycle of Machine learning Framework



Taxanomy of Machine Learning

The diverse ML algorithms can be disaggregated at the topmost level into two categories namely Supervised Learning Algorithms and Unsupervised Learning Algorithms.

Supervised Learning Algorithms : This category of algorithms is used to either numerically predict a target feature by mapping its relationship with the other features or generating a categorical classification of the target feature in context of the other features. This classification generates classes with levels.

Supervised Algorithms are provided explicit instructions as to what it needs to learn and how the algorithm will learn. The algorithm optimizes the model to find the best combination of features to obtain the target feature.

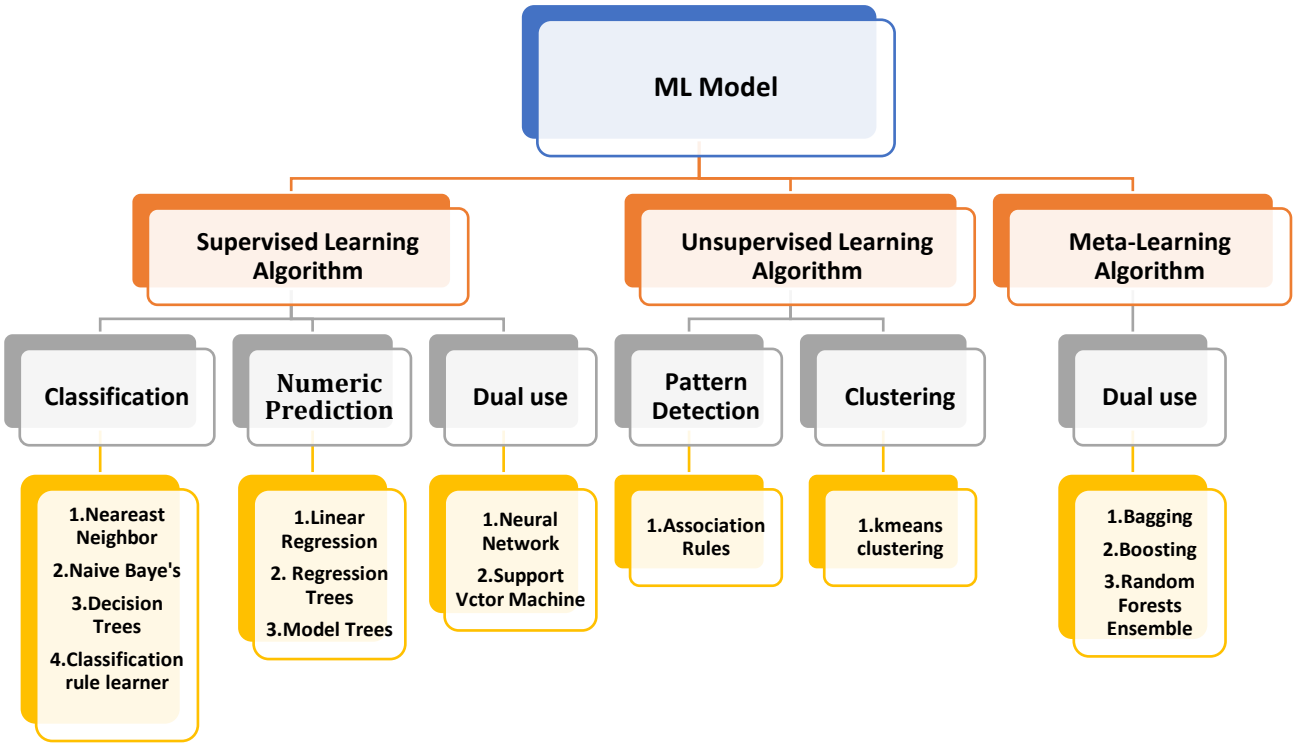
Examples prediction of risk of cancer patients, spam email prediction.

Unsupervised Learning Algorithms

Is the identification of the inherent pattern in a data set. The pattern discovery for the unsupervised learning does not entail having a target feature , on the contrary it is training a descriptive model with no feature more important than the other. For instances if we wish to ascertain the shopping patterns of individuals people who buy soda might also buy chips. Gamblers , thieves might have the same habits. The differentiation of descriptive model into homogeneous groups creates clusters which have to be interpreted and analyzed by data scientists. This segmentation procedure is called Clustering.

Meta Learners

These are algorithms that learning how to learn more effectively. This optimizes the final prediction.



The basic strategy of implementing a machine learning algorithm is identifying from amongst four learning tasks namely Classification, Numeric prediction, Pattern recognition and clustering. The more nuanced and detailed oriented review will be performed in subsequent lectures.

Class Work

1) Install R , Install R Studio
2) Download the data from the book
3) Create a github account
4) Choose data set from Kaggle, create a project group