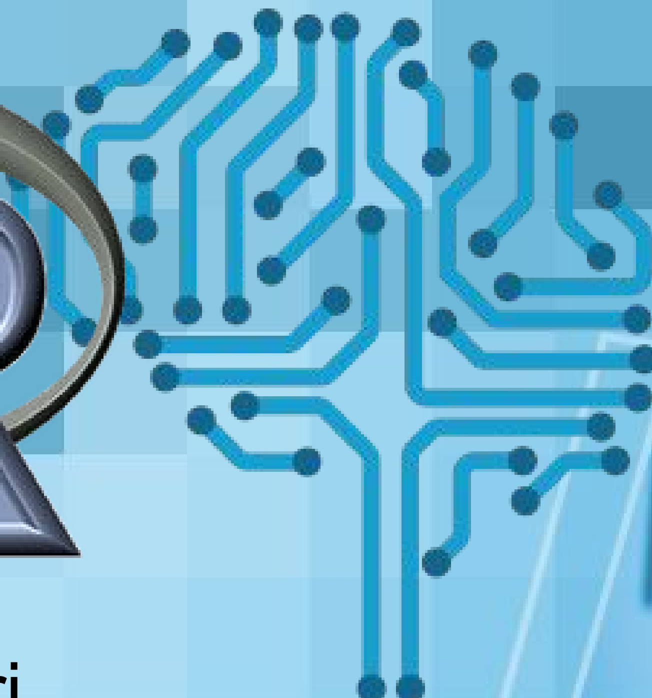
R Machine Learning



Trainer: Dr. Ravi Tiwari

Website:www.tertiarycourses.com.sg
Email: enquiry@tertiaryinfotech.com

About the Trainer

Dr. Ravi Kumar Tiwari got his PhD from NUS (Chemical Engineering) in 2013. After graduation, he worked 3 years as a research scientist in the Institute of High Performance Computing (IHPC). He is currently a data scientist at Fujitsu. His core skills are R, big data, Hadoop and machine learning.

Agenda

Module 1 Introduction to Machine Learning

- What is Machine Learning
- R packages for ML
- Installing R ML packages

Module 2 Datasets

- Datasets for MM
- Features
- Iris Dataset
- Boston Housing Price Dataset
- Mtcars Dataset
- Splitting Datasets for Training/Testing

Agenda

Module 3 Supervised Learning

- What is Supervised Learning
- Metric
- Decision Tree Classifier
- Random Forest Classifier
- KNN Classifier
- KNN Regression
- Linear Regression (Ridge and Lasso Regularization)
- Logistics Regression Classifier
- SVM Classifier
- GNB Classifier

Agenda

Module 4 Unsupervised Learning

- What is Unsupervised Learning
- Clustering
- Dimensionality Reduction

Module 5 Intro to Neural Network (Optional)

- What is Neural Network
- Multi Layer Perceptron

Prerequisite

Basic knowledge of R is assumed

Exercise Files

ynload the exercise file from

ps://github.com/rkrtiwari/rMachi
Learning

Module 1
Getting Started

What is Machine Learning?

- Machine Learning is about building programs with tunable parameters that are adjusted automatically so as to improve their behavior by adapting to previously seen data
- Machine Learning is a subfield of Artificial Intelligence

Why Machine Learning?

http://www.goratings.org/

Rank	Name	♂♀	Flag	Elo
1	Google DeepMind AlphaGo			3608
2	Ke Jie	♂		3608
3	Park Junghwan	♂		3593
4	Lee Sedol	♂		3550
5	Iyama Yuta	♂		3536
6	Mi Yuting	♂		3528

Machine Learning

- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
  - Clustering
  - Dimensionality Reduction

R Packages for ML

- rpart
- randomForest
- e1071
- glmnet
- nnet
- class
- FNN

Installing and Loading R ML Packages

```
install.packages("rpart")
library(rpart)
```

Module 2
Datasets

Iris Flower Dataset

Iris Flower Dataset

setosa (0) versicolor (1) virginica (2)

Iris flower dataset, introduced in 1936 by Sir Ronald Fisher

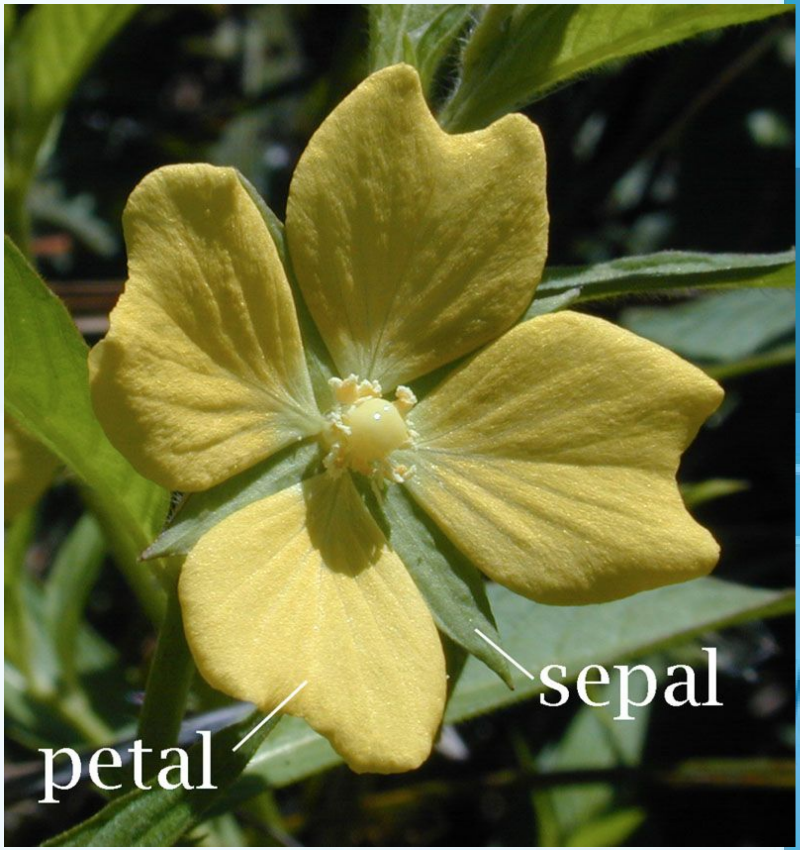
Iris Flower Dataset

Features in the Iris dataset:

- sepal length in cm
- sepal width in cm
- petal length in cm
- petal width in cm

Target classes to predict:

- setosa
- versicolor
- virginica



Load Iris Dataset

```
data(iris)
dim(iris)
levels(iris$Species)
head(iris)
```

Boston Housing Price Dataset

Boston Housing Price Dataset

There are 13 features for this dataset.

- CRIM per capita crime rate by town
- ZN proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS proportion of non-retail business acres per town
- CHAS Charles River dummy variable
- NOX nitric oxides concentration (parts per 10 million)
- RM average number of rooms per dwelling
- AGE proportion of owner-occupied units built prior to 1940
- DIS weighted distances to five Boston employment centres
- RAD index of accessibility to radial highways
- TAX full-value property-tax rate per $10,000
- PTRATIO pupil-teacher ratio by town
- B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- LSTAT % lower status of the population
- MEDV Median value of owner-occupied homes in $1000's

Load Boston Housing Dataset

```
library(MASS)
Boston
dim(Boston)
head(Boston)
```

Mtcars Dataset

Motor Trend Car (mtcars) dataset

There are 11 features for this dataset.

- mpg Miles/(US) gallon
- cyl Number of cylinders
- disp Displacement (cu.in.)
- hp Gross horsepower
- drat Rear axle ratio
- wt Weight (lb/1000)
- qsec 1/4 mile time
- vs V/S
- am Transmission (0 = automatic, 1 = manual)
- gear Number of forward gears
- carb Number of carburetors

Load MTCars Dataset

mtcars

dim(mtcars)

head(mtcars)

Splitting Datasets for Training/Testing

Splitting Dataset for Testing

```
index <- sample(c(TRUE, FALSE), n,
                replace = TRUE, prob = c(0.6,0.4))
```

n is the # of rows in dataset

```
train <- iris[index,]
test <- iris[!index,]
```

Module 3
Supervised Learning

What is Supervised Learning

- In Supervised Learning, we have a dataset consisting of both features and labels.
- The input data (X) is associated with a target label (y)

Supervised Learning Examples

- Spam Email Filter
- Tumor Classification

Classification Steps

```
# Step 1 Load classifer library
library(package)

# Step 2 Split the data
index <- sample(....prob = c(0.6, 0.4))

# Step 3 Training
model <- classifier(y ~ ., data = train)

# Step 4: Prediction
class <- predict(model, data = test)
```

Decision Tree Classifier

Load the library

```
library(rpart)
```

Split the Iris Dataset

```
index <- sample(c(TRUE, FALSE), nrows(iris),
replace = TRUE, prob = c(0.6, 0.4))


train <- iris[index, ]
test <- iris[!index,]
```

Build the tree model

```
model <- rpart(Species ~ ., data = train)
```

Make Prediction

class <- predict(model, newdata = test, type = “class”)

Verify Model Prediction

```
mean(class == test[,5])   # Accuracy
table(class, test[,5])    # Confusion Matrix
```

Ex: Decision Tree Classifier

Use Decision Tree regressor to build a model to predict media house price (MEDV) using boston dataset

Time: 5 mins

Random Forest Classifier

Load the library

```
library(randomForest)
```

Split the data set

```
index <- sample(c(TRUE, FALSE), replace = TRUE,
prob = c(0.6, 0.4))

train <- iris[index, ]
test <- iris[!index,]
```

Build the Random Forest model

```
model <- randomForest(Species ~ ., data = train, mtry = 3, ntree=20)
```

Make Prediction

```
class <- predict(model, newdata = test, type = “class”)
```

Assess Model Prediction

```
mean(class == test[,5])   # Accuracy
table(class, test[,5])    # Confusion Matrix
```

Challenge

Use random forest regressor to build a model to predict media house price (MEDV) using boston dataset

Time: 5 mins

K-Nearest Neighbour

Load the library

```
library(class)  # For classification
library(FNN)  # For regression
```

Split the data set

```
index <- sample(c(TRUE, FALSE), nrow(iris),
            replace = TRUE, prob = c(0.6, 0.4))

train <- iris[index, ]
test <- iris[-index,]
```

Make Prediction

```
class <- knn(train[,1:4], test[,1:4],
                  y = train[,5], k = 3)
```

Assess Model Prediction

```
mean(class == test[,5])   # Accuracy
table(class, test[,5])     # Confusion Matrix
```

Challenge

Use knn to build a model to predict media house price (MEDV) using boston dataset

Time: 5 mins

Linear Regression

Build the linear regression model

```
model <- lm(mpg ~ wt, data = mtcars)
```

Make Prediction

```
value <- predict(model, data.frame(wt = mtcars$wt))
```

Access the model parameters

```
coef(model)
sumModel <- summary(model)
sumModel$r.squared
```

Multivariate linear regression

```
model <- lm(mpg ~ ., data = mtcars)
```

Challenge

Make a linear regression model to predict the median house price using boston data set. Find the RMS error of the model

Time: 5 mins

Regularization

Load the library

```
library(glmnet)
```

CV to determine the best penalty parameter using Lasso

```
model <- cv.glmnet(x,y, alpha=1, nfolds = 5)
bestlam  <- model$lambda.min
```

*alpha = 0 gives ridge regression

Prediction at best penalty parameter

```
value <- predict(model ,s=bestlam ,newx=x1)
```

Logistic Regression

Split the data set

```
index <- sample(c(TRUE, FALSE), nrow(iris),
                replace = TRUE, prob = c(0.6, 0.4))

train <- iris[train, ]
test <- iris[!train,]
```

Build the Logistic Regression model

```
model <- glm(Species ~ Petal.Length, data = train,
family = binomial(link="logit"))
```

Make Prediction

```
prob <- predict(model, newdata = test,
                type = "response")
class <- ifelse(prob > 0.5, "virginica",
                "versicolor")
```

Access Model Prediction

```
mean(class == test[,5])  # Accuracy
table(class, test[5])        # Confusion Matrix
```

Support Vector Machine

Load the library

```
library(e1071)
```

Split the data set

```
index <- sample(c(TRUE, FALSE), nrow(iris),
                replace = TRUE, prob = c(0.6, 0.4))

train <- iris[index, ]
test <- iris[!index,]
```

Build the SVM model

```
model <- svm(Species ~ ., data = train,
             kernal = "linear", scale = TRUE)

model <- svm(Species ~ ., data = train,
               kernal = "radial", scale = TRUE,
               cost = 1, gamma = 0.5)
```

Make Prediction

```
class <- predict(model, newdata = test)
```

Assess Model Prediction

```
mean(class == test[,5])  # Accuracy
table(class,test[,5])        # Confusion Matrix
```

Challenge

Use svm to build a model to classify a flower species using it sepal and petal measurements

Time: 5 mins

Gaussian Naive Bayes

Load the library

```
library(e1071)
```

Split the data set

```
index <- sample(c(TRUE, FALSE), nrow(iris),
          replace = TRUE, prob = c(0.6, 0.4))

train <- iris[index, ]
test <- iris[!index,]
```

Build the GNB Model

```
model <- naiveBayes(Species ~ Petal.Length,
                    data = train)
```

Make Prediction

```
class <- predict(model, test)
```

Assess Model Prediction

```
mean(class == test[,5])  # Accuracy
table(class,test[,5])         # Confusion Matrix
```

Module 4
Unsupervised Learning

Clustering

Hierarchical Clustering

```
m <- dist(iris)
hc <- hclust(m)
clusters <- cutree(hc, k = 3)
```

Challenge

Using hierarchical clustering find 3 clusters in the mtcars dataset. Do not include mpg variable for clustering.

Hint: Scale the data before clustering

Time: 5 min

k-means Clustering

```
kmeans(iris, centers = 3, nstart = 10)
```

Challenge

1. Using kmeans clustering find 3 clusters in the mtcars dataset. Do not include mpg variable for clustering.

Hint: Scale the data before clustering

Time: 5 min

Dimensionality Reduction

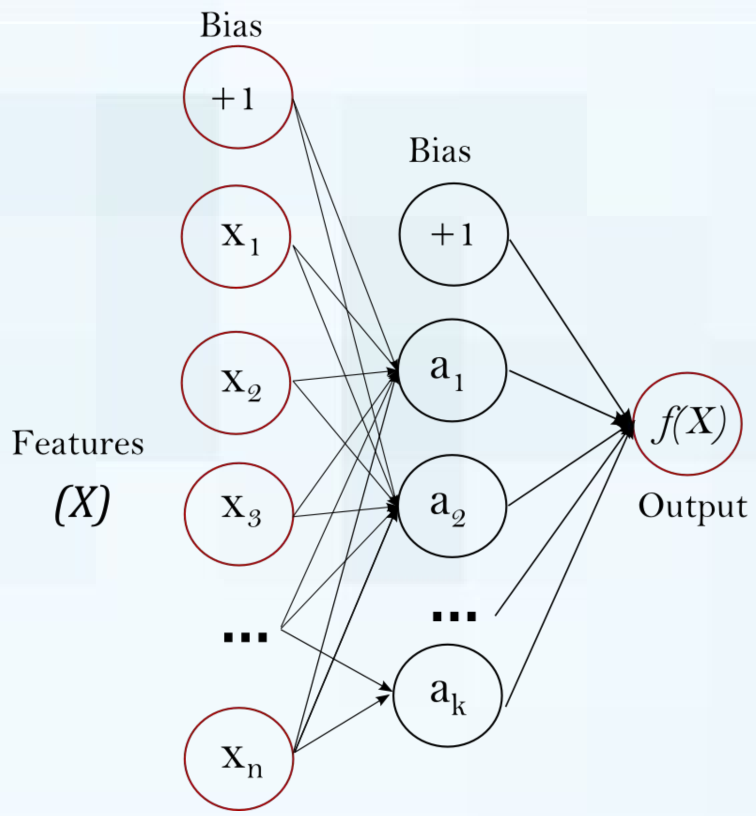
```
pcl <- prcomp(iris[,1:4], scale = TRUE)
pcl$rotation
pcv <- pcl$sdev^2; pve <- pcv/sum(pcv)



plot(pve, xlab = "Principal Component", ylab =
  "Proportion of variance explained", ylim=c(0,1),
  type="b")
```

Module 5
Neural Network
(Optional)

One Layer MLP



Load the library

```
library(nnet)
```

Split the data set

```
index <- sample(c(TRUE, FALSE), nrow(iris),
                replace = TRUE, prob = c(0.6, 0.4))

train <- mtcars[index, ]
test <- mtcars[!index,]
```

Build the neural network model

```
model <- nnet(mpg ~ ., data = train, size = 3,
              linout = TRUE, skip = TRUE)
```

Make Prediction

```
value <-  predict(model, test)
```

Assess Model Prediction

```
mean((value - test[,1])^2)
```

Challenge

Make a neural network model to predict the median house price using boston data set. Find the RMS error of the model

Practice Makes Perfect

Summary

Parting Message

Q&A
Feedback

https://www.tertiarycourses.com.sg/course-feedback.html

Thank You!

Dr. Ravi Kumar Tiwari
9119 6694
rkrtiwari@gmail.com