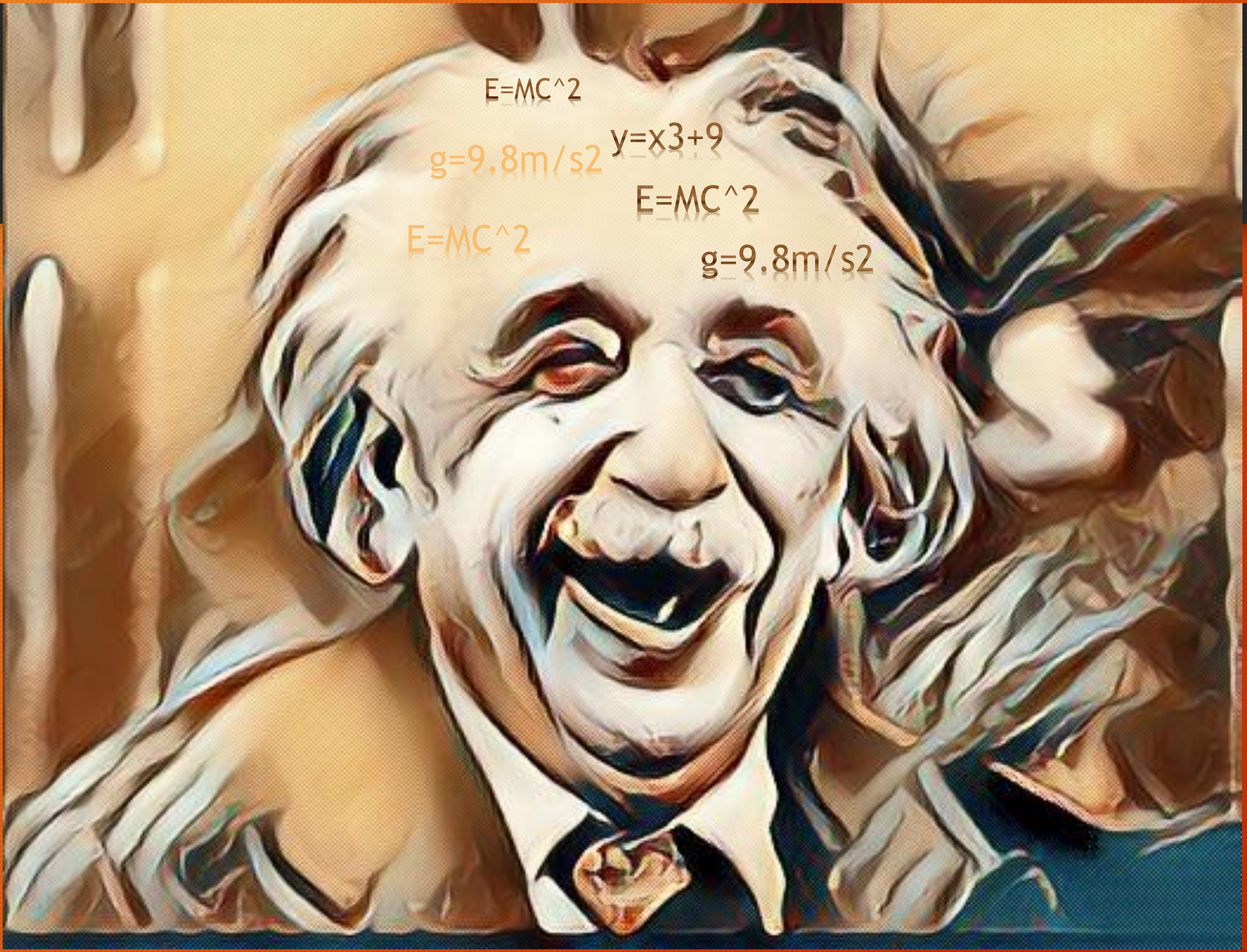
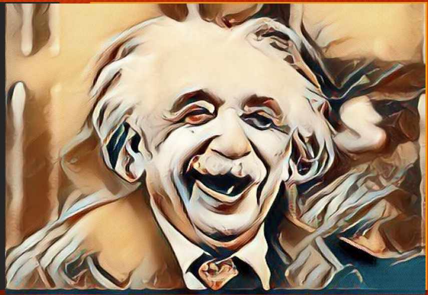


Data Science

Machine Learning - Do it yourself

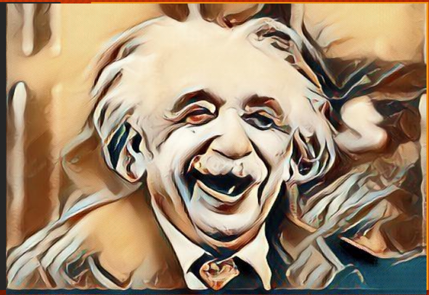
by
Bharati DW Consultancy
cell: +1-562-646-6746 (Cell & Whatsapp)
email: bharati.dwconsultancy@gmail.com
website: http://bharaticonsultancy.in/

Features of the course



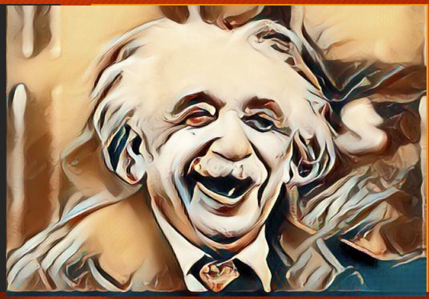
- ✓ Course is oriented towards Data Science – No prerequisites.
- ✓ Primary goal is to give a kick-start to a Data Science Career.
- ✓ Follow the examples mentioned in the videos.
- ✓ Each topic is accompanied by Hands-on exercises.
- ✓ Deep focus on real-life time examples.
- ✓ Assuming no background experience – learn it at your own-pace.
- ✓ For every section you will find video links to understand the concepts better.

Agenda



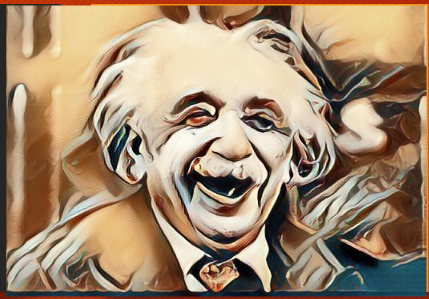
- ✓ Introduction to Machine Learning
- ✓ Crash course on R, Data Structures, & frequently used commands.
- ✓ Supervised & Unsupervised Machine Learning Algorithms.
- ✓ Hands-on on real time examples
- ✓ Data Science real-life project.

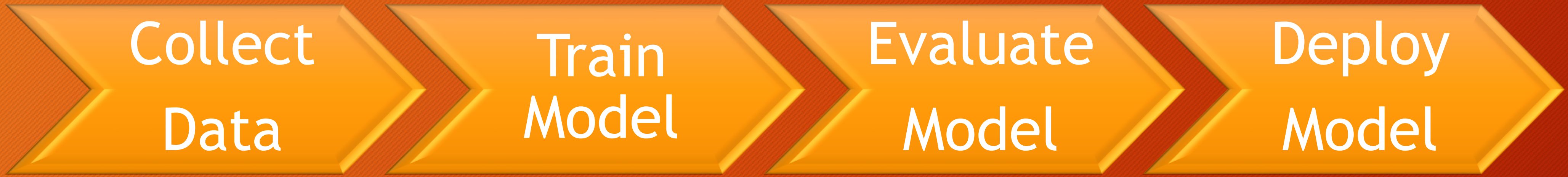
Machine Learning – an Intro



- ✓ Type of AI which provides an ability to learn without explicit programming – by feeding them data.
- ✓ Data is the fuel for machine learning, which looks for patterns in the data, to be able to develop a program called as a model.
- ✓ When exposed to new data, this computer model is enabled to learn, change and grow.

Machine Learning – an Intro (contd.)





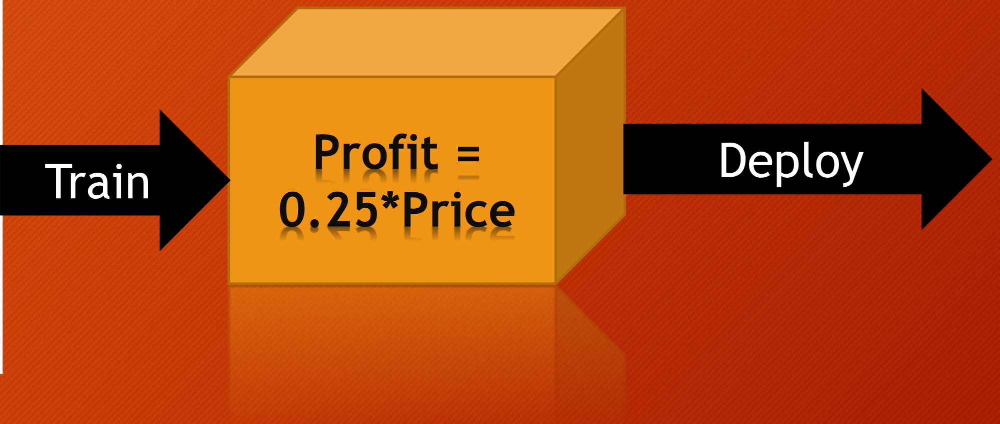
Collect Data

Train Model

Evaluate Model

Deploy Model

Price	Profit
4	1
16	4
20	5
100	25

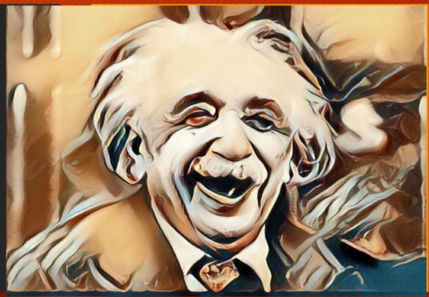


Train

Profit = 0.25*Price

Deploy

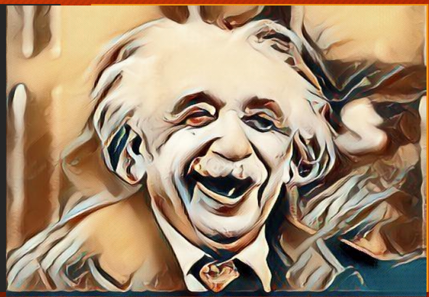
Types of Machine Learning Algorithms



- Supervised Learning
  - ✓ Model is trained to predictive a Target feature.
  - ✓ Classification or Numeric Predictions

- Unsupervised Learning
  - ✓ Descriptive Modeling task – pattern discovery.
  - ✓ Segmentation Analysis
  - ✓ Pattern detection or Clustering

Machine Learning Algorithms



Supervised

Classification

> Nearest Neighbor
> Naïve Bayes
> Neural Networks
> Decision Tree
> Classification Rules
> Support Vectors

Numeric Predictions

> Linear Regression
> Regression Trees
> Neural Networks
> Support Vectors

Unsupervised

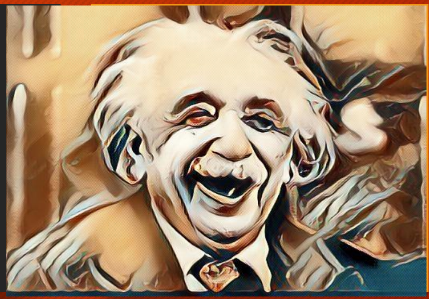
Pattern Detection

> Association Rules

Clustering

> k-means Clustering

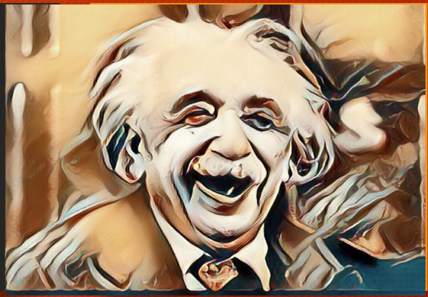
R for Machine Learning



- ✓ Download and install R from CRAN – Comprehensive R Archive Network
- ✓ Precompiled binary distributions of the base system and contributed packages.
- ✓ Most common Machine Learning language
- ✓ Can be integrated with other tools like Tableau

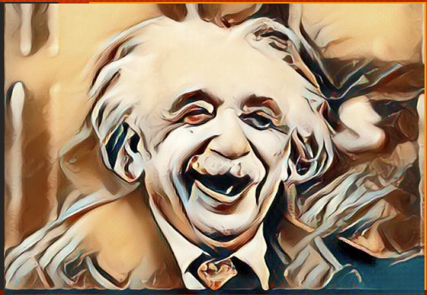
https://cran.r-project.org/index.html

Hands On – R Machine Learning Ex-1



- ✓ Download and install R.
- ✓ Check the libraries installed – use library()

Video

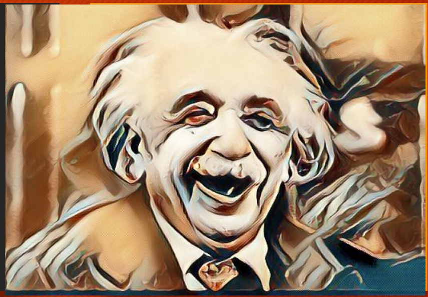


✓ For every section you will find video links to understand the concepts better.

✓ Getting Started - DIY- 1 -of-50

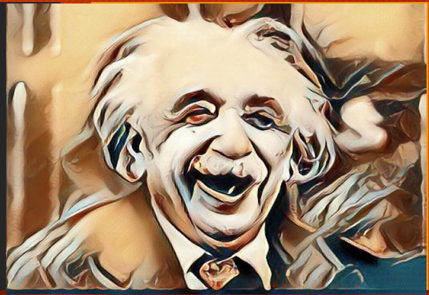
https://www.youtube.com/watch?v=sc4WxfJFvh4

Data Structures in “R”



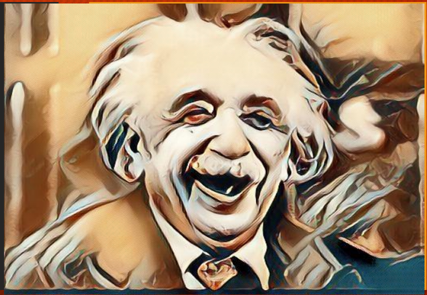
- ✓ Vectors
- ✓ Factors
- ✓ Lists
- ✓ Matrices
- ✓ Data Frames

Vectors



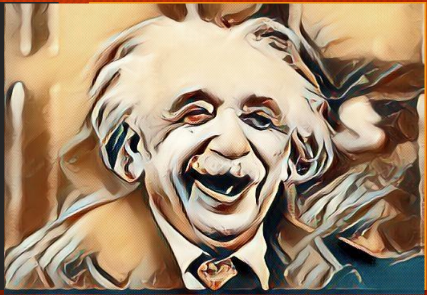
- ✓ Collection of data elements.
- ✓ Elements should be of the same Data type. Use *typeof(<vector>)*
- ✓ *produce <- c("Banana","Banana Peppers","Apple","Pineapple")*
- ✓ *inventory <- c(8,4,40,50)*
- ✓ *Refill <- c(TRUE,TRUE,FALSE,FALSE)*

Hands On – R Machine Learning Ex-2



- ✓ Create the following vectors.
  - ❖ *cust_nm* with elements- David, John, Jeff, Steve, Swati, Nikki, Lisa
  - ❖ *gender* with elements- Male,Male,Male,Male,Female,Female,Female
  - ❖ age with elements- 20,25,44,45,24,28,36
  - ❖ Income with elements – 65,75,80,110,70,65,90
  - ❖ SalesAmt with elements – 100, 110,125,150,105,95,140
- ✓ Display the elements and datatype of these vectors.

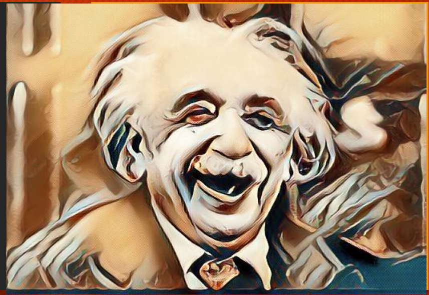
Factors



- ✓ Features that represent a characteristic with categories of values - MALE/FEMALE, Fruit/Veg/Meat etc. - nominals.
- ✓ Helps reduce the memory size
- ✓ *Produce_Type <- factor(c("Fruit", "Vegetable","Fruit", "Vegetable"))*
- ✓ *Produce_Type1 <- factor(c("Fruit","Vegetable")*

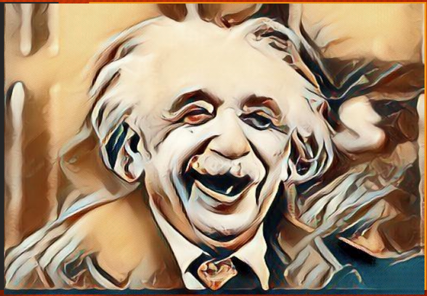
, levels = c("Fruit","Vegetable","Meat"), ordered=TRUE <FALSE>)

Hands On – R Machine Learning Ex-3



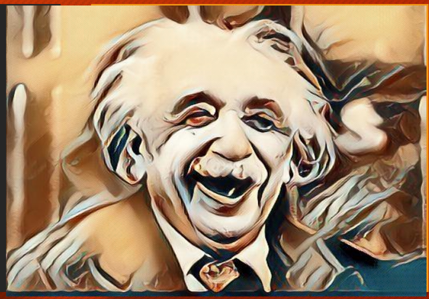
- ✓ Create the following Factors.
  - ❖ *gender* with elements- Male,Male,Male,Male,Female,Female,Female
  - ❖ employed with elements – self-emp, emp,emp,emp,self-emp,emp,emp
  - ❖ Marital_status with elements – single,single,married, married,single,single,married – with possible levels of single, married & divorced.
- ✓ Display these Factors.

Videos



- ✓ R Data Structures - DIY- 2 -of-50
- ✓ https://www.youtube.com/watch?v=Ht8O5JW9GrI
- ✓ R Data Structures - Factors - DIY- 3 -of-50
- ✓ https://www.youtube.com/watch?v=6tgUbEBJnJA

Lists



✓ Similar to Vectors – they are collection of ordered set of data elements.

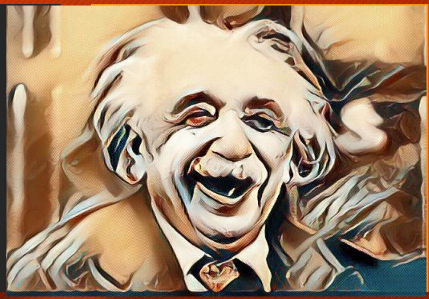
✓ Elements may not be of the same Data type.

✓ *MyList1 <- list(produce = produce[1], inventory = inventory[1], Produce_Type = Produce_Type[1],*

Refill = Refill [1])

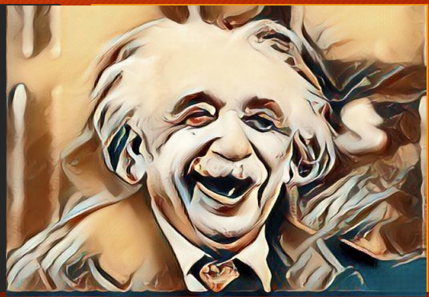
✓ *MyList <- list(produce,Inventory, Produce_Type,Refill)*

Matrices



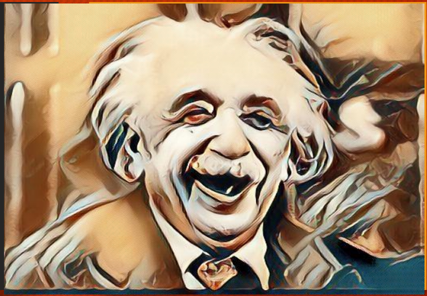
- ✓ Has both rows and columns.
- ✓ Contains data with same data types like vectors.
- ✓ Creating a matrix needs a vector with number of rows or columns.
- ✓ *MyMatrix1 <- matrix(c(1,2,3,4,5,6),nrow=2)*
- ✓ *MyMatrix2 <- matrix(c(1,2,3,4,5,6),ncol=2)*

Hands On – R Machine Learning Ex-4



- ✓ Create a list using – MyList_cust with all the customer vectors and factors created in ex – 2 & 3.
- ✓ Display the list features.
- ✓ Create matrices each for 4 rows, 4 columns, 5 rows and 5 columns.

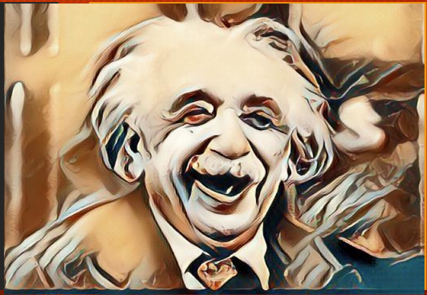
Videos



- ✓ R Data Structures - List & Matrices - DIY- 4 -of-50

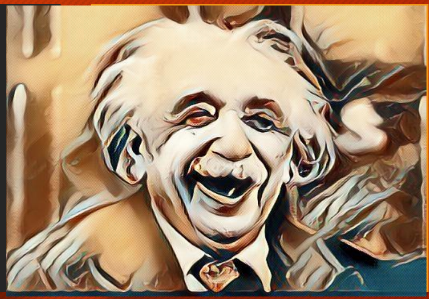
  https://www.youtube.com/watch?v=GApMimnormc

Data Frames



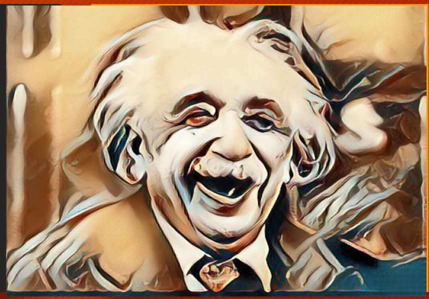
- ✓ Has both rows and columns. Analogous to excel spreadsheet or database.
- ✓ List of vectors or factors.
- ✓ stringsAsFactors = FALSE. Else R will automatically convert every character vector to a factor.
- ✓ *MyDataFrame <- data.frame(produce,inventory, Produce_Type, Refill, stringsAsFactors=FALSE)*

Data Frames (contd…)



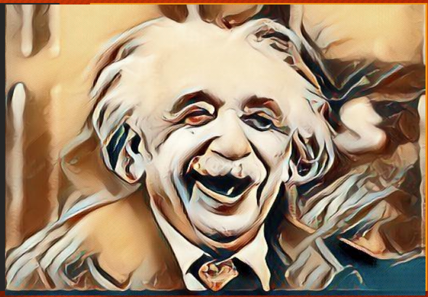
- ✓ *MyDataFrame$produce*
- ✓ *MyDataFrame[c(“produce”,”inventory”)]*
- ✓ *MyDataFrame[2:4]* - shows column 2 to 4.
- ✓ *MyDataFrame[1,2]* - intersection of matrix cell 1 & 2.
- ✓ *MyDataFrame[c(1,2), c(2,4)]* - shows 1st & 2nd rows for columns 2 & 4.
- ✓ *MyDataFrame[1,]* - shows all columns for for the first row.
- ✓ *MyDataFrame[,1]* - shows all rows for the first column.
- ✓ *MyDataFrame[,]* - shows all rows & columns.

Hands On – R Machine Learning Ex-5



- ✓ Create a DataFrame – MyDF_cust with all the customer vectors and factors created in ex – 2 & 3.
- ✓ Display these DataFrame features.

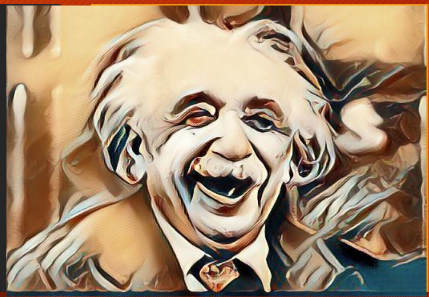
Videos



- ✓ R Data Structures - Data Frames - DIY- 5 -of-50

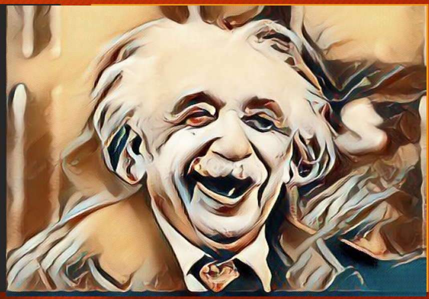
https://www.youtube.com/watch?v=8SEOtWRZ91k

Frequently used R - Commands



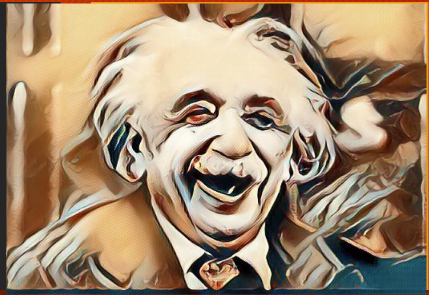
- ✓ *ls()* - Lists the Data Structures.
- ✓ *save(MyDataFrame,MyMatrix, file="c:\\R\\MyData.RData")*
- ✓ *rm(MyDataFrame)* - removes the MyDataFrame.
- ✓ *load("c:\\R\\MyData.RData")* - loads the stored RData.
- ✓ *rm(list=ls())* - removes all the data structures.
- ✓ *str(MyDF_cust)* - is like row count(*) & describe table columns with datatypes
- ✓ *summary(MyDF_cust$age)* - displays several common summary statistics
- ✓ summary(*MyDF_cust*[c("age","SalesAmt")])
- ✓ mean(*MyDF_cust$age*)

Hands On – R Machine Learning Ex-6



- ✓ Save all of your data structures created in the earlier hands on exercises onto a location in c:\R\ folder.
- ✓ Find summary & means from Income group, age & SalesAmt from the data structures loaded from the saved file.

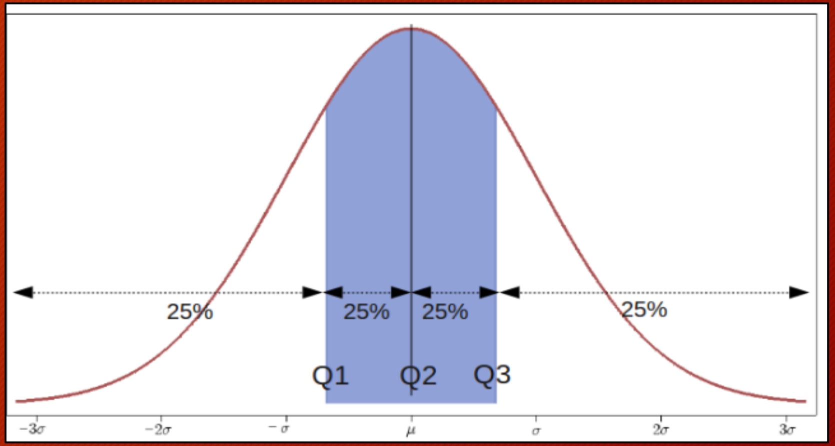
Frequently used R – Commands (contd…)



✓ *range(MyDF_cust$Age)* - range of age.

✓ diff(*range(MyDF_cust$SalesAmt)*)

✓ quantile(*MyDF_cust$SalesAmt)* - quantiles are cut points dividing the range of a probability distribution into contiguous intervals with equal probabilities, or dividing the observations in a sample in the same way.



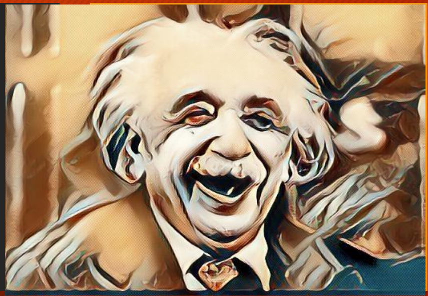
✓ quantile(*MyDF_cust$SalesAmt, probs = c(0.1,0.99))*

✓ quantile(*MyDF_cust$SalesAmt, probs = c(0.01,0.99))*

✓ quantile(*MyDF_cust$SalesAmt, probs = c(0.0001,0.999999))*

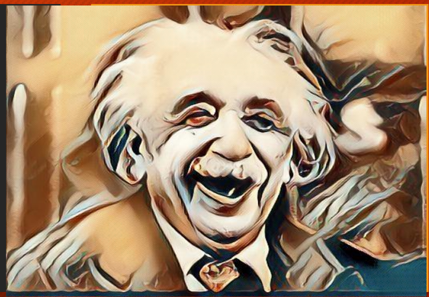
✓ quantile(*MyDF_cust$SalesAmt, seq(from = 0, to = 1, by = 0.15))*

Frequently used R – Commands (contd…)



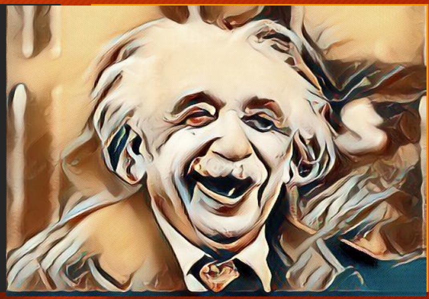
- ✓ *Seq(n)* – where n can be a number.
  - *seq(stats::rnorm(20)) or seq(1, 9, by = 2) or seq(0, 1, length.out = 11)*
- ✓ *var(MyDF_cust$SalesAmt)* – variance
- ✓ *sd(MyDF_cust$SalesAmt)* – standard deviation
- ✓ *table(MyDF_cust$Gender)* – displays the distribution of the Gender.
- ✓ *prop.table(table(MyDF_cust$Gender))* – shows proportion from the table Gender.
- ✓ my_table <- *table(MyDF_cust$Gender)*
- ✓ *prop.table(my_table)*

Hands On – R Machine Learning Ex-7



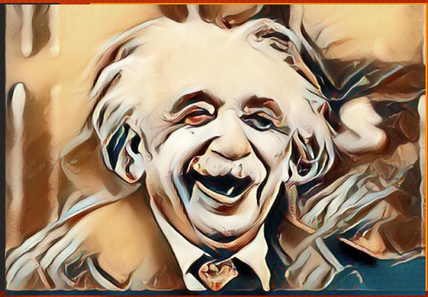
- ✓ Display range, quantile, variance and standard deviation for Income and age from the MyDF_Cust dataframe.
- ✓ Display range, quantile, variance and standard deviation for Inventory & price from the MyDataFrame.
- ✓ Display table for Marital_status, proportion from the MyDF_cust dataframe created earlier.

R – Commands for handling CSVs



- ✓ loadmycsv <- read.csv("c:\\R\\Prod_inv.csv",stringsAsFactors = FALSE);
- ✓ loadmycsv
- ✓ loadmycsv1 <- read.csv("c:\\R\\Prod_inv.csv",stringsAsFactors = FALSE,header=FALSE);
- ✓ write.csv(loadmycsv,file = "c:\\R\\Prod_invtest.csv",row.names=FALSE);

Videos



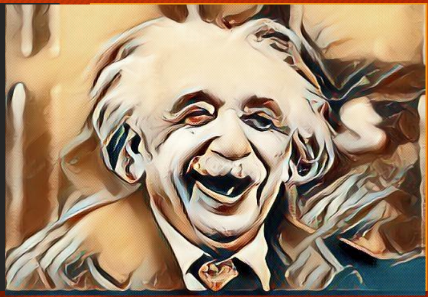
- ✓ Frequently used R commands - DIY- 6 -of-50

  https://www.youtube.com/watch?v=FaWpgQVYMMU

- ✓ Frequently used R commands contd - DIY- 7 -of-50

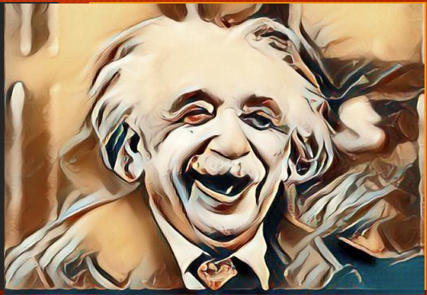
  https://www.youtube.com/watch?v=ES-1_zLYcRE

Installing RStudio



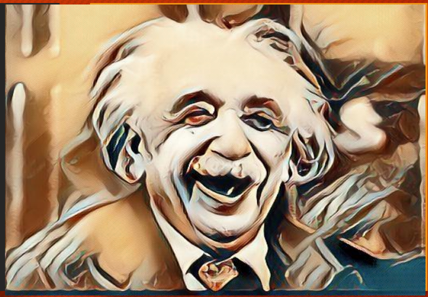
- ✓ Provides IDE for R.
- ✓ Go to https://www.rstudio.com/
- ✓ Click on Download
- ✓ Choose the RStudio Desktop Version
- ✓ Select the appropriate version.
- ✓ Install RStudio from the downloaded file.

R – Data Visualization – boxplot



- ✓ The boxplot() function – displays the center & spread of a numeric variable.
- ✓ Shows the spread of the data, with median denoted by a solid black line.
- ✓ *boxplot(loadmycsv$Inventory)*
- ✓ *boxplot(loadmycsv$Inventory, main="My Chart")*
- ✓ *boxplot(loadmycsv$Inventory, main="My Chart",ylab="Inventory (x10 lbs)")*
- ✓ *boxplot(loadmycsv$Price, main="My Chart",ylab="Price ($$)")*
- ✓ *boxplot(csv$Price, main="My Chart",ylab="Price ($$)", xlab="My x axis")*

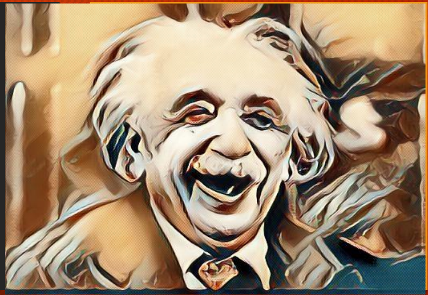
Videos



- ✓ Installing RStudio- DIY- 8 -of-50

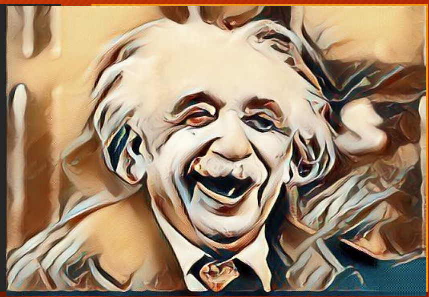
  https://www.youtube.com/watch?v=82Uo73d_JXc

R – Data Visualization – Histogram



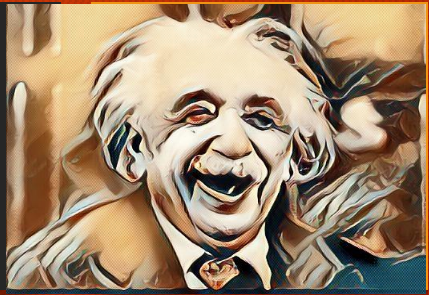
- ✓ The hist() function – displays the frequency of occurrence of a numeric value.
- ✓ Shows the data in groups called as Bins.
- ✓ *hist(loadmycsv$Inventory)*
- ✓ *hist(loadmycsv$Inventory, main="My Chart")*
- ✓ *hist(loadmycsv$Inventory, main="My Chart",xlab="Inventory (x10 lbs)")*
- ✓ *hist(loadmycsv$Price, main="My Chart",ylab="Price ($$)")*

R – Data Visualization – plot/Scatterplot.



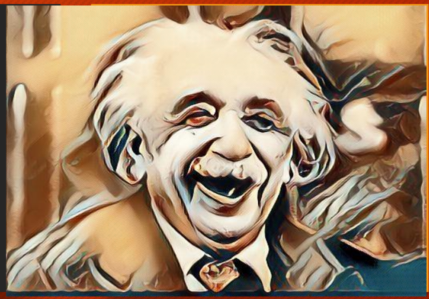
- ✓ The plot() function – plots a 2 dimensional graph.
- ✓ A Scatterplot shows bivariate relationship.
- ✓ *plot(x=MyDF_cust$age, y=MyDF_cust$SalesAmt, xlab = “Age",ylab=“Income")*

Hands On – R Machine Learning Ex-8



- ✓ Read the Customer_Age_Income.csv file in a dataframe MyDF_CAI.
- ✓ Calculate standard deviation on the age.
- ✓ Save the MyDF_cust dataframe as MyDF_cust.csv
- ✓ Create Graphs for boxplot, histogram, and scatterplot for MyDF_CAI – age, income, SalesAmt.

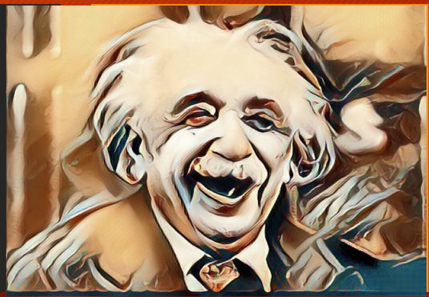
Videos



- ✓ R Data Visualization Basics - DIY- 9 -of-50

  https://www.youtube.com/watch?v=72alsyrW3dU

Machine Learning Algorithms – Recap



Supervised

Classification

- > Nearest Neighbor
- > Naïve Bayes
- > Neural Networks
- > Decision Tree
- > Classification Rules
- > Support Vectors

Numeric Predictions

- > Linear Regression
- > Regression Trees
- > Neural Networks
- > Support Vectors

Unsupervised

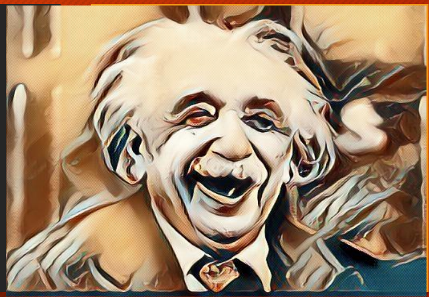
Pattern Detection

- > Association Rules

Clustering

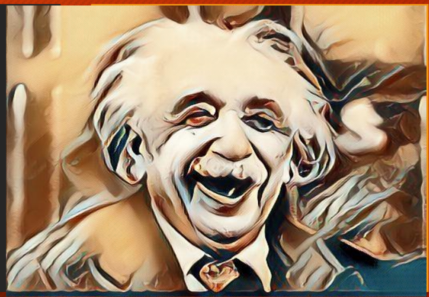
- > k-means Clustering

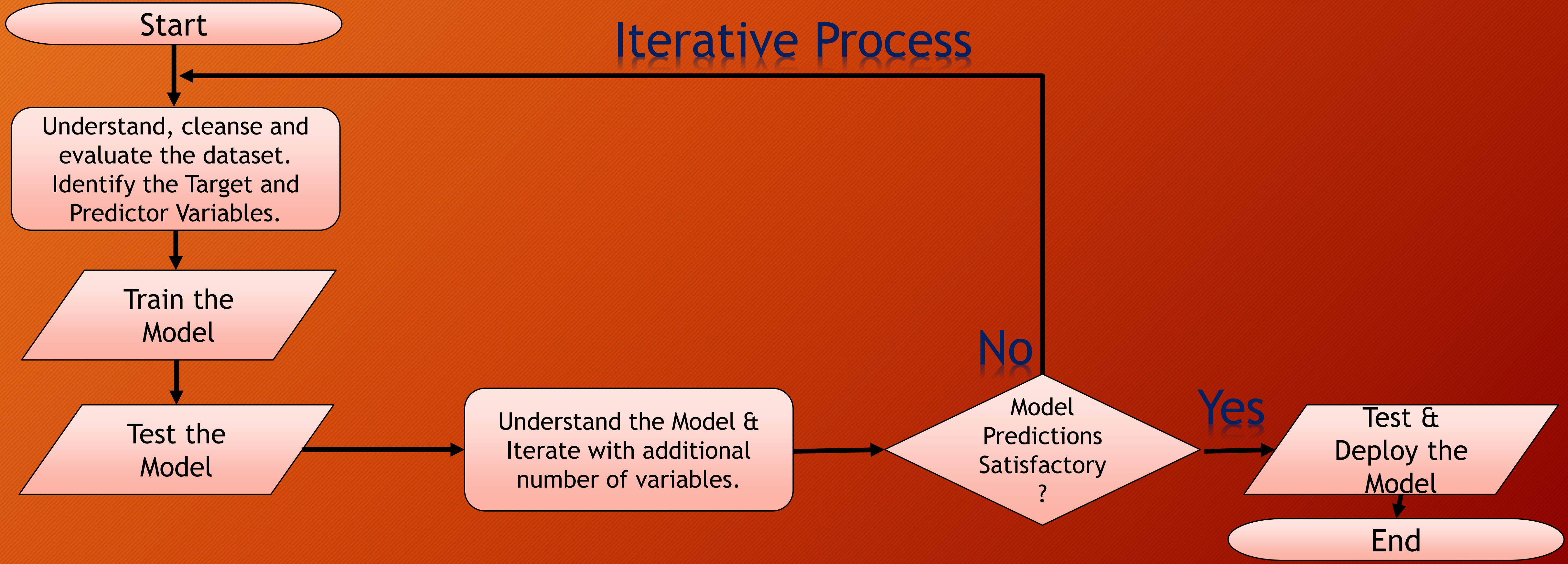
Linear Regression Models



- ✓ Simple linear regression is a statistical method.
- ✓ It allows summarization and study of relationships between two continuous (quantitative) variables.
- ✓ Attempts to explore, **model** the relationship between two or more variables.
- ✓ With more than one explanatory variable, the process is called Multiple Linear Regression.

Linear Regression Model (Contd.)





Start

Understand, cleanse and evaluate the dataset. Identify the Target and Predictor Variables.

Train the Model

Test the Model

Understand the Model & Iterate with additional number of variables.

Model Predictions Satisfactory ?

No

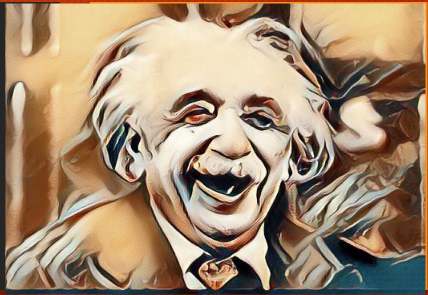
Iterative Process

Yes

Test & Deploy the Model

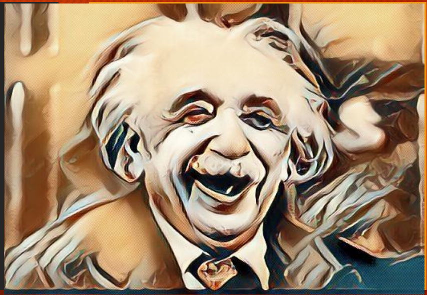
End

Hands On – R Machine Learning Ex-9



- ✓ Download the data Customer_Age_Income.csv from the google drive link.
- ✓ Implement Simple Linear Regression Model for target variable - Spend using predictor variables Income & then Job.
- ✓ What happens when the target variable is Income and predictor variable is Spend.

Videos



- ✓ Linear Regression Model - DIY- 10(a) -of-50

  https://www.youtube.com/watch?v=zQkRSKcanIU

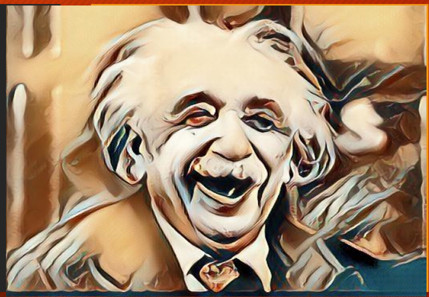
- ✓ Linear Regression Model - DIY- 10(b) -of-50

  https://www.youtube.com/watch?v=3KVco80HRnE

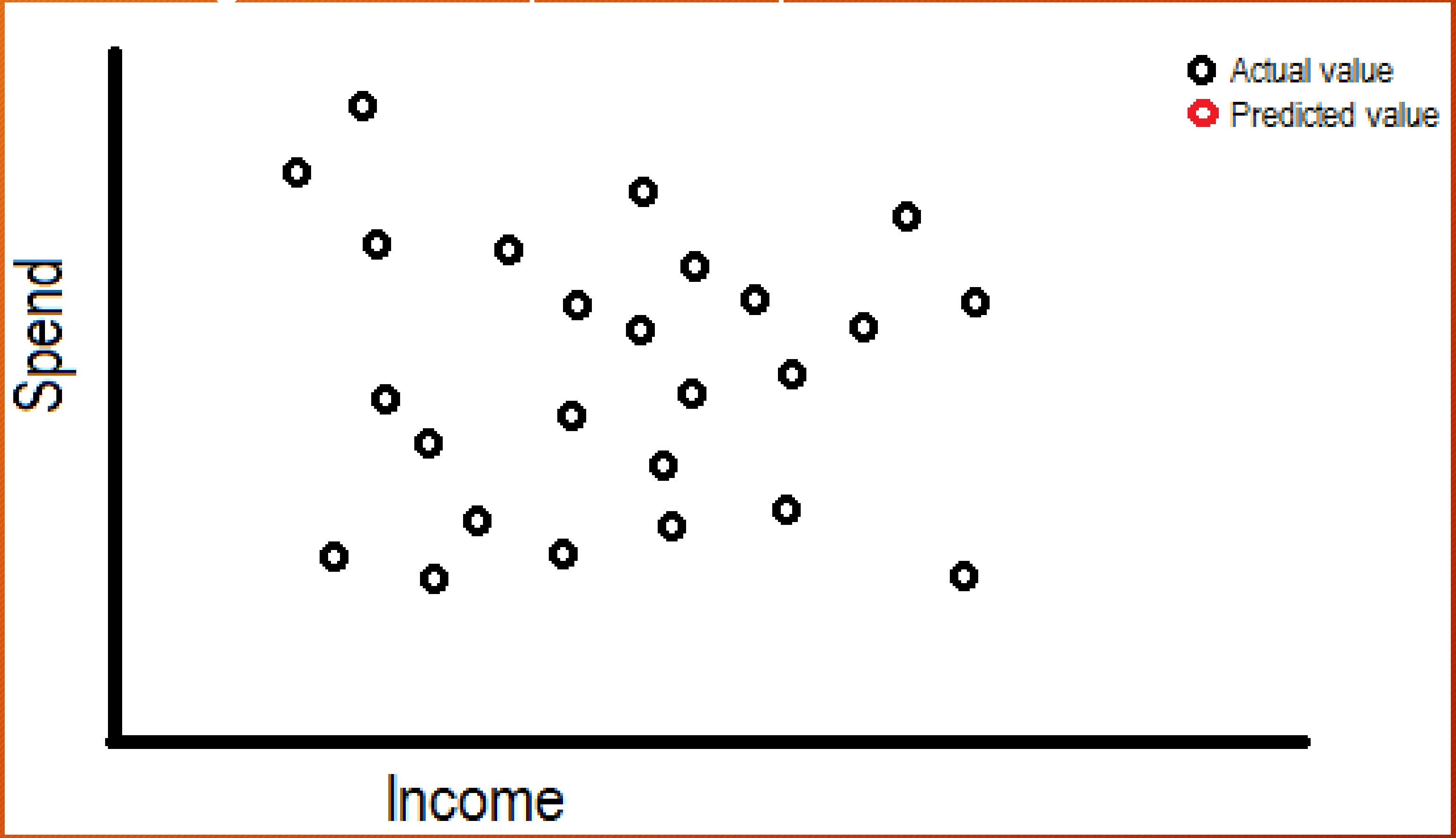
- ✓ Multiple Linear Regression Model - DIY- 11 -of-50

  https://www.youtube.com/watch?v=orD_H8avz0g

Evaluating Model Performance

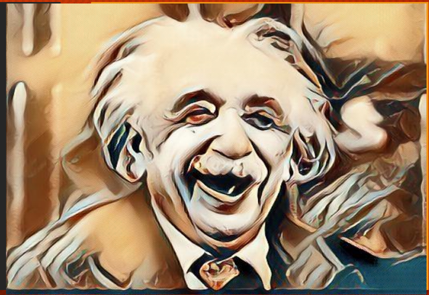


Linear Regression – Simple & Multiple Actual vs Predicted Values

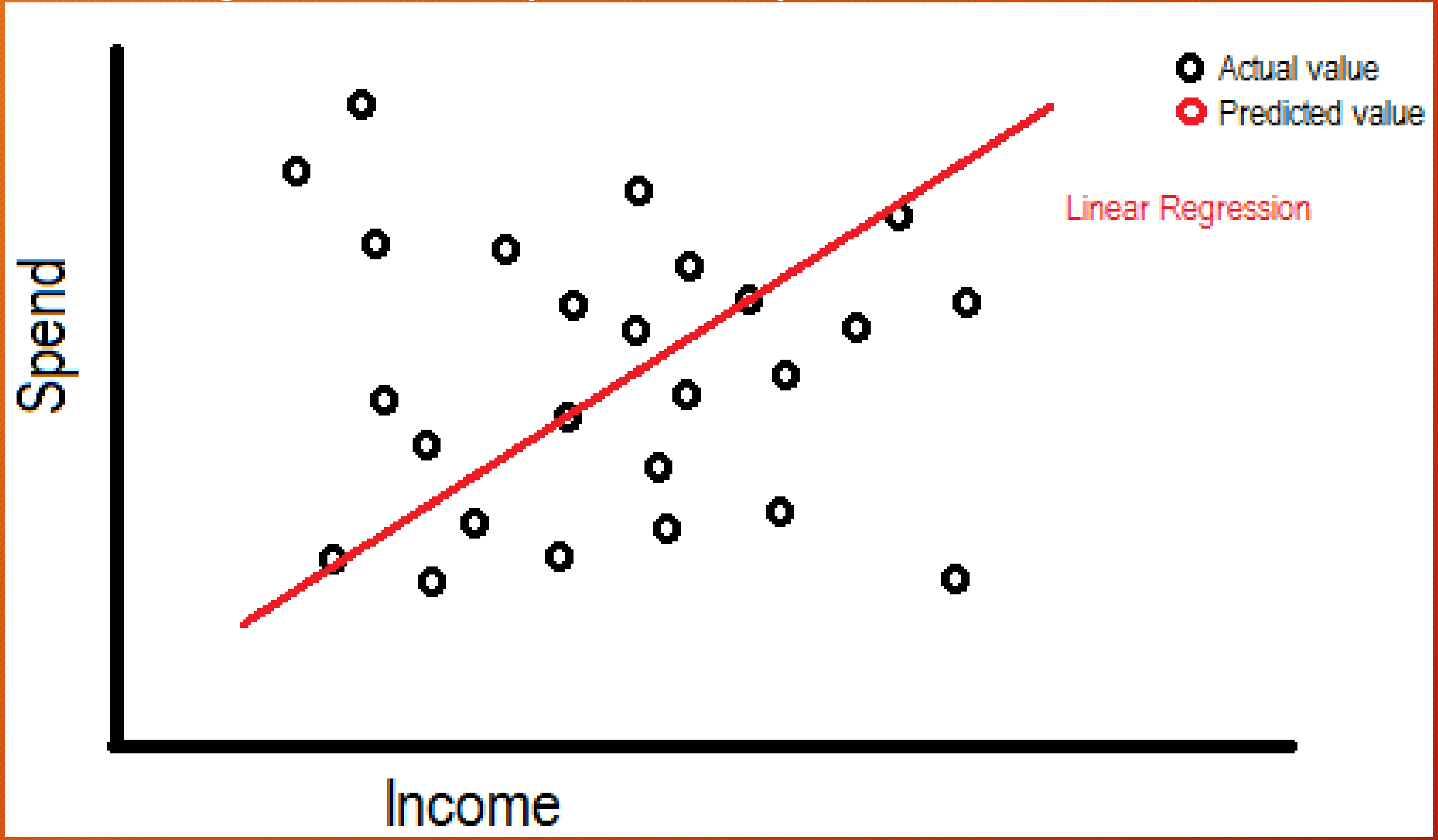


- ✓ Plot the Actual values

Evaluating Model Performance



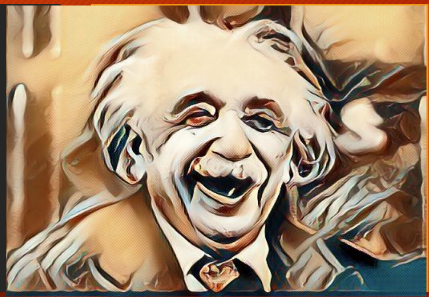
Linear Regression – Simple & Multiple Actual vs Predicted Values



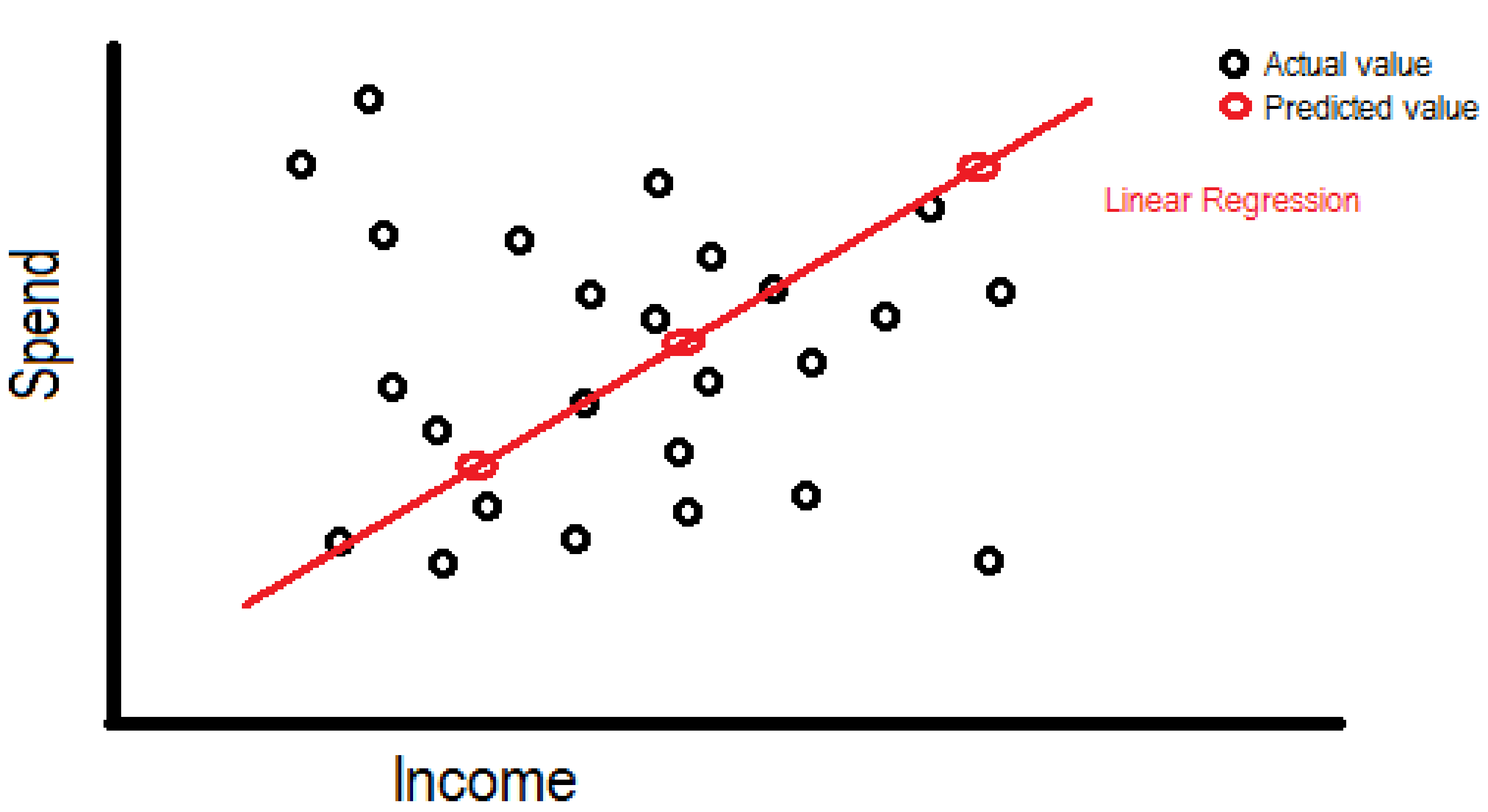
- ✓ Plot the Actual values
- ✓ Draw a Line
- ✓ Equation:

  Slope y = mx + b

Evaluating Model Performance

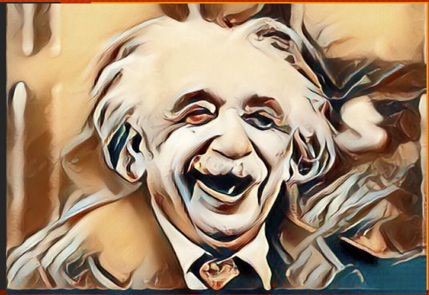


Linear Regression – Simple & Multiple Actual vs Predicted Values



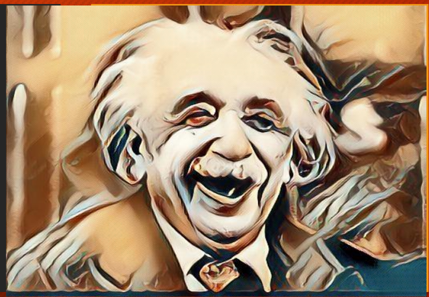
- ✓ Plot the Actual values
- ✓ Draw a Line
- ✓ Equation:
  Slope y = mx + b
- ✓ KPI -
- ✓ Root-mean-square deviation (RMSD) or root-mean-square error (RMSE)
- ✓ R-Squared

RMSE / RMSD



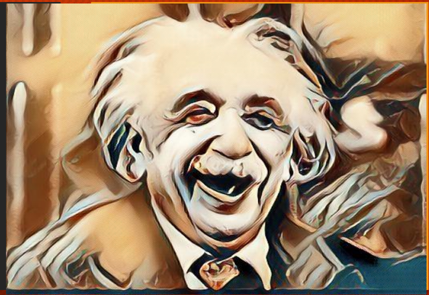
- ✓ The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) is a very commonly used measure of the differences between predicted values by a model and the actual values seen in the data.
- ✓ The square root of the mean/average of the square of all of the error.
- ✓ It compares the forecasting errors of various models for a target variable.
- ✓ RMSE <- sqrt(mean((predicted – actual) ^2));

R-Squared



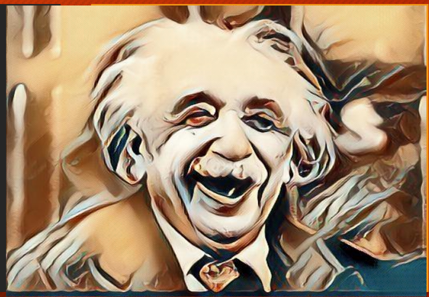
- ✓ R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple predicted values.
- ✓ R-squared is conveniently scaled between 0 and 1, whereas RMSE is not scaled to any particular values.
- ✓ You would want R-Squared closer to 1.

Hands On – R Machine Learning Ex-10



- ✓ Download the data Customer_Age_Income.csv from the google drive link.
- ✓ Implement Simple & Multiple Linear Regression Model for target variable - Spend using predictor variables Age, Income, Job, Auto Loan Indicator, Gender, Marital Status.
- ✓ Note RMSE & R-Squared for each additional variable added.

Videos



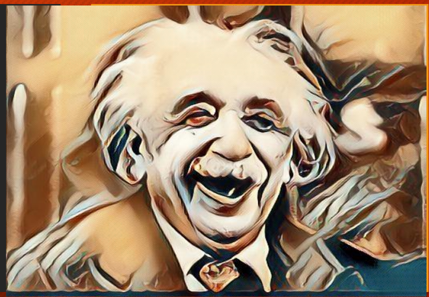
- ✓ Evaluate Model Performance - DIY- 12 -of-50

  https://www.youtube.com/watch?v=uMeQc_crlIc

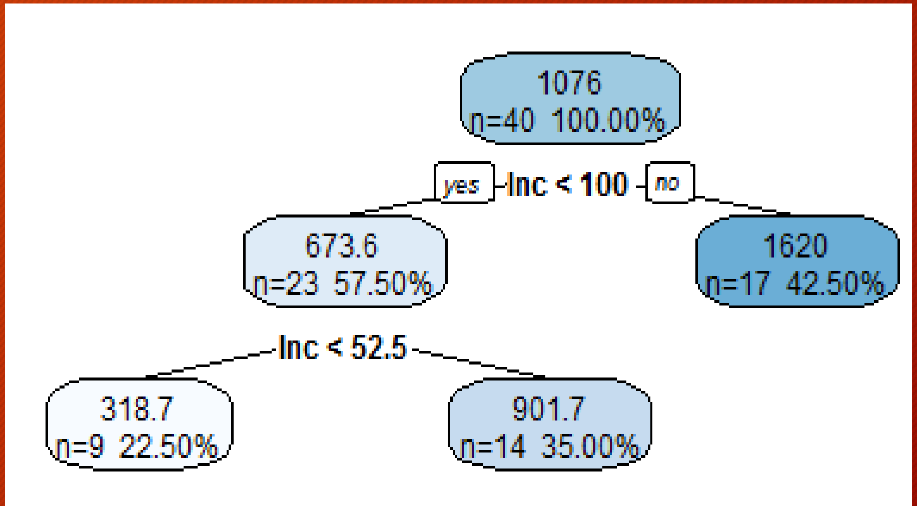
- ✓ RMSE & R-Squared - DIY- 13 -of-50

  https://www.youtube.com/watch?v=pYFqFUGObts

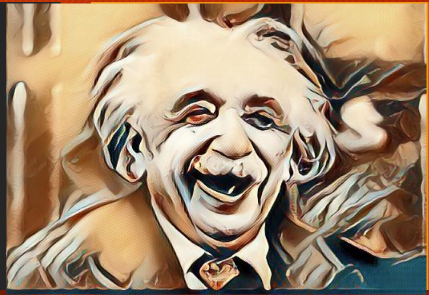
Regression Trees



- ✓ Regression trees are similar to Flowchart / Decision tree.
- ✓ A tree consists of decision nodes, leaf nodes.
- ✓ A Sample Regression Tree output.

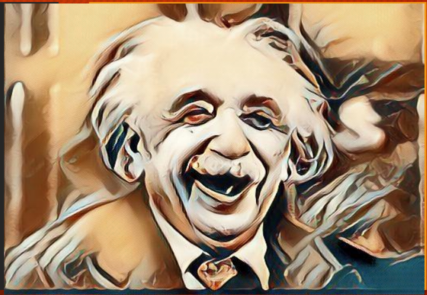


Regression Trees (contd.)



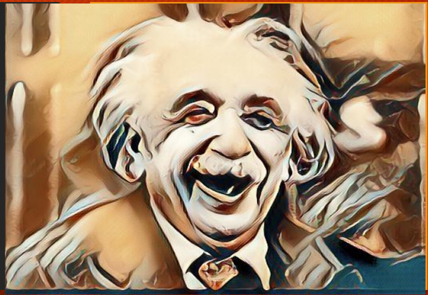
- ✓ Decision Trees are generally used for Classification.
- ✓ Regression Trees may also be used for Numeric Predictions.
- ✓ They bring together the ability of decision trees to model and predict numeric data.
- ✓ They may make predictions using the average values of examples that reach a leaf, and (or) creating a Linear model using the examples reaching a node.
- ✓ They may fit some types of data much better than Linear Regression

Hands On – R Machine Learning Ex-11



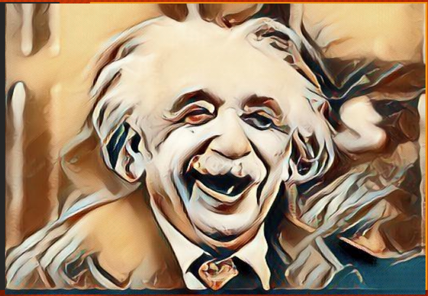
- ✓ Download the data Customer_Age_Income.csv from the google drive link.
- ✓ Implement Regression Trees Model for target variable - Spend using predictor variables Age, Income, Job, Auto Loan Indicator, Gender, Marital Status.

Regression Trees (*Methods*)



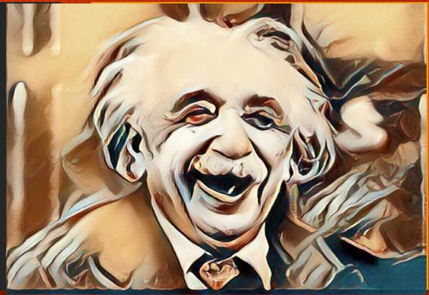
- ✓ *rpart*(Target ~ Predictors, *data* = DataSets, *method*="Types").
- ✓ Method Types:
  - ❖ *class* – categorical classification of data.
  - ❖ *anova* – continuous values.
  - ❖ *poisson* – based on counts of values, like count of employed
  - ❖ *exp* – exponential - Survival method

Hands On – R Machine Learning Ex-12



- ✓ Extend the hands-on exercise -11
- ✓ Implement Regression Trees Model using different methods for target variable - Spend using predictor variables Age, Income, Job, Auto Loan Indicator, Gender, Marital Status.
- ✓ Calculate Mean Square Error for each method.

Videos



- ✓ Numeric Predictions using Regression Trees - DIY- 14 -of-50

  https://www.youtube.com/watch?v=Np1wXBdizvl

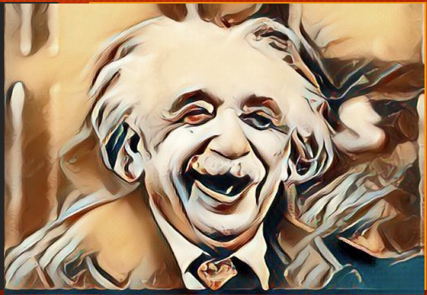
- ✓ Regression Decision Trees contd - DIY- 15 -of-50

  https://www.youtube.com/watch?v=mt0Dv_2izgo

- ✓ Method Types in Regression Trees - DIY- 16 -of-50

  https://www.youtube.com/watch?v=qO9_RQgnSGA

Hands On – R Machine Learning – Real time Project-1



- ✓ Time for a real-time project.
- ✓ Download data files from the following URL.

  http://archive.ics.uci.edu/ml/datasets/Energy+efficiency#

- ✓ Let’s understand the dataset.

Citation:

A. Tsanas, A. Xifara: 'Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools', Energy and Buildings, Vol. 49, pp. 560-567, 2012

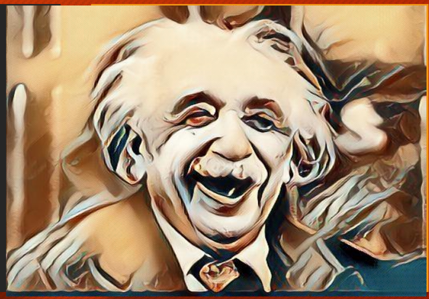
(the paper can be accessed from [Web Link])

For further details on the data analysis methodology:

A. Tsanas, 'Accurate telemonitoring of Parkinson’s disease symptom severity using nonlinear speech signal processing and statistical machine learning', D.Phil. thesis, University of Oxford, 2012

(which can be accessed from [Web Link])

Hands On – R Machine Learning – Real time Project-1

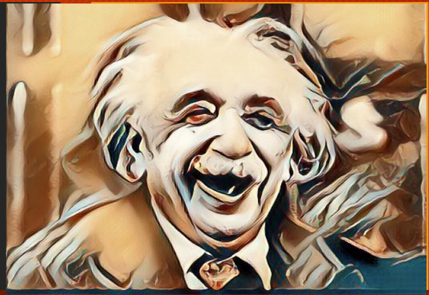


✓ Data elements:

❖ Specifically:

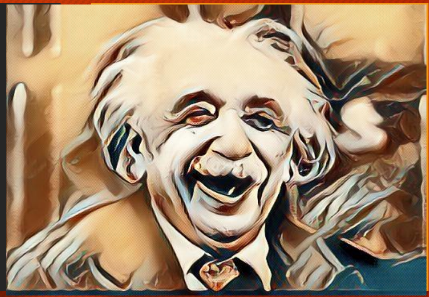
X1	Relative Compactness
X2	Surface Area
X3	Wall Area
X4	Roof Area
X5	Overall Height
X6	Orientation
X7	Glazing Area
X8	Glazing Area Distribution
y1	Heating Load
y2	Cooling Load

Hands On – R Machine Learning – Real time Project-1



- ✓ Load this data in any database – MYSQL, Oracle, DB2 etc.
- ✓ For your convenience, the data set has already been loaded in a Microsoft Access File, available for to you download from our google drive link - *Energy_efficiency.mdb*
- ✓ Implement Linear Regression Model & Regression Trees Model for target variables – Heating Load and Cooling Load.

Videos



- ✓ Real Time Project 1 - DIY- 17 -of-50

  https://www.youtube.com/watch?v=AzCN-BFu-cE

- ✓ Helping to create solution to Real Time Project- DIY- 18 -of-50

  https://www.youtube.com/watch?v=yMQJMf9uEZs

K – Nearest Neighbors (K-NN)

- Get the data from Balance Scale Data Set.
- Attribute Information:
  - Class Name: 3 (L, B, R)
  - Left-Weight: 5 (1, 2, 3, 4, 5)
  - Left-Distance: 5 (1, 2, 3, 4, 5)
  - Right-Weight: 5 (1, 2, 3, 4, 5)
  - Right-Distance: 5 (1, 2, 3, 4, 5)
- http://archive.ics.uci.edu/ml/datasets/Balance+Scale

Citation Policy:

If you publish material based on databases obtained from this repository, then, in your acknowledgements, please note the assistance you received by using this repository. This will help others to obtain the same data sets and replicate your experiments. We suggest the following pseudo-APA reference format for referring to this repository:

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

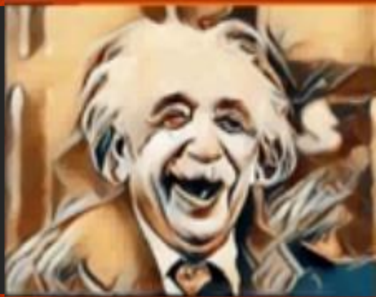
Here is a BiBTeX citation as well:

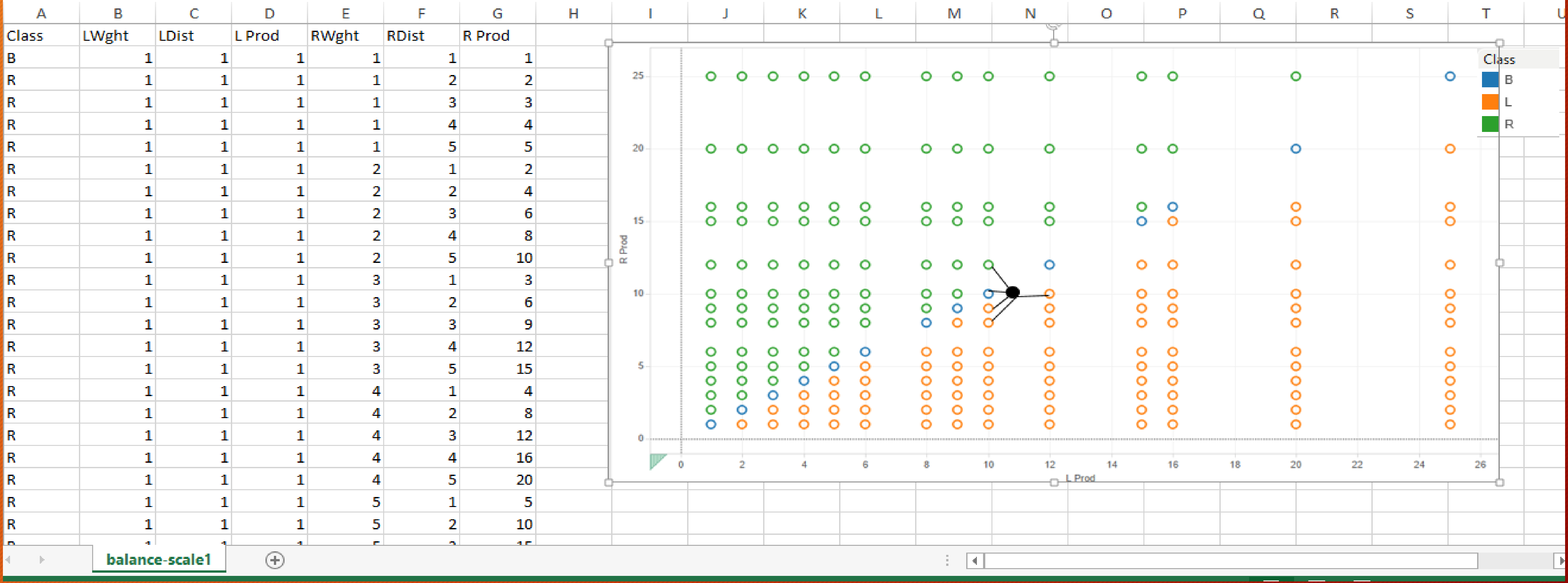
@misc{Lichman:2013 , author = "M. Lichman",

year = "2013", title = "{UCI} Machine Learning Repository",

url = "http://archive.ics.uci.edu/ml", institution = "University of California, Irvine, School of Information and Computer Sciences" }

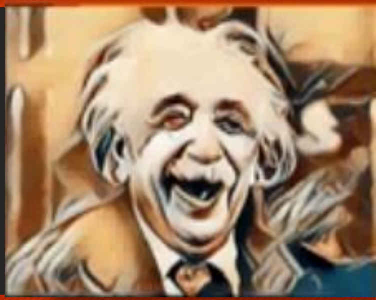
K – Nearest Neighbors





Class	LWght	LDist	L Prod	RWght	RDist	R Prod
B	1	1	1	1	1	1
R	1	1	1	1	2	2
R	1	1	1	1	3	3
R	1	1	1	1	4	4
R	1	1	1	1	5	5
R	1	1	1	2	1	2
R	1	1	1	2	2	4
R	1	1	1	2	3	6
R	1	1	1	2	4	8
R	1	1	1	2	5	10
R	1	1	1	3	1	3
R	1	1	1	3	2	6
R	1	1	1	3	3	9
R	1	1	1	3	4	12
R	1	1	1	3	5	15
R	1	1	1	4	1	4
R	1	1	1	4	2	8
R	1	1	1	4	3	12
R	1	1	1	4	4	16
R	1	1	1	4	5	20
R	1	1	1	5	1	5
R	1	1	1	5	2	10

K – Nearest Neighbors



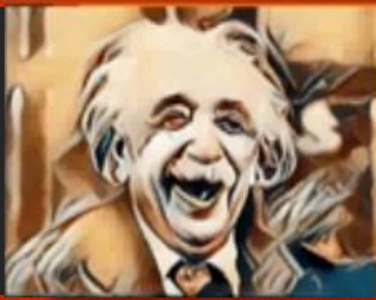
- The nearest neighbors is a very simple and effective approach to classification which are well-suited for classification tasks.

- k-NN utilizes information about an example's k-nearest neighbors to classify unlabeled examples.

- k- generally is an odd-number for nearest neighbors – in case of tie-breaker.

- k neighbors in the training data that are the "nearest" in similarity. The unlabeled test instance is then assigned the class of the majority of the k nearest neighbors.

K – Nearest Neighbors



- k-NN algorithm uses Euclidean distance, which is the distance one would measure if it were possible to use a ruler to connect two points.

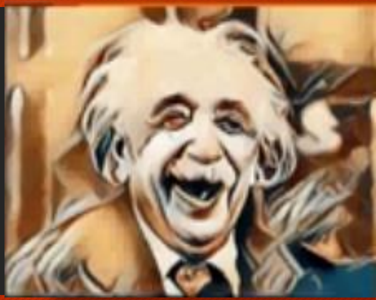
- Choosing k- common practice is to begin with k equal to the square root of the number of training sample.

- *knn(train* = <train_data>, *test* = <test_data>, *cl* = <train_labels>, *k* = <num>*)*

- Use normalize function if the range of values are really

  normalize <- function(y) {return ((y - min(y)) / (max(y) - min(y)))}

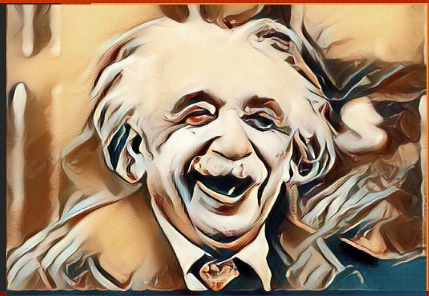
Hands On – R Machine Learning Ex-13



- Download the dataset k-NN algorithm, it uses Euclidean distance, which is the distance one would measure if it were possible to use a ruler to connect two points.
- http://archive.ics.uci.edu/ml/datasets/Glass+Identification
- Choose k- equal to the square root of the number of training sample.
- *knn(train* = <train_data>, *test* = <test_data>, *cl* = <train_labels>, *k* = <num>*)*
- You may have to use the normalize function.

  normalize <- function(y) {return ((y - min(y)) / (max(y) - min(y)))}

Videos



- ✓ KNN Classification - DIY- 21 -of-50

  https://www.youtube.com/watch?v=AC0j_rtFZRg

- ✓ KNN Classification Hands on - DIY- 22 -of-50

  https://www.youtube.com/watch?v=S9Hv3KRrnyU

- ✓ KNN Classification Hands on Contd - DIY- 23 -of-50

  https://www.youtube.com/watch?v=S94y56-iLws

- ✓ KNN Classification Exercise - DIY- 24 -of-50

  https://www.youtube.com/watch?v=s_h4EzqO_Eo

C5.0 Decision Tree - Classification

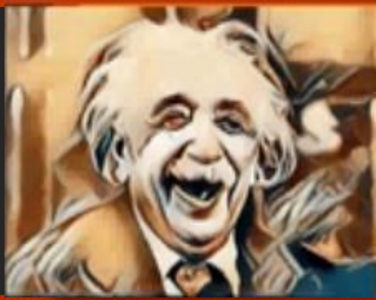
- Decision trees are very powerful classifiers, which utilize a tree structure to model the relationships among the features and the potential outcomes.

- An all-purpose classifier which has a highly automatic learning process; it can handle numeric or nominal features.

- C5.0 uses entropy, a concept analogous to the information theory that quantifies the randomness, or disorder, within a set of class values.

```
C50_model<- C5.0(train_Predictors, train_Target)
C50_predict<- predict(C50_model, test_data)
```

C5.0 Decision Tree - Classification



- Get the data from Balance Scale Data Set.
- Attribute Information:
  - Class Name: 3 (L, B, R)
  - Left-Weight: 5 (1, 2, 3, 4, 5)
  - Left-Distance: 5 (1, 2, 3, 4, 5)
  - Right-Weight: 5 (1, 2, 3, 4, 5)
  - Right-Distance: 5 (1, 2, 3, 4, 5)
- http://archive.ics.uci.edu/ml/datasets/Balance+Scale

Citation Policy:

If you publish material based on databases obtained from this repository, then, in your acknowledgements, please note the assistance you received by using this repository. This will help others to obtain the same data sets and replicate your experiments. We suggest the following pseudo-APA reference format for referring to this repository:

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

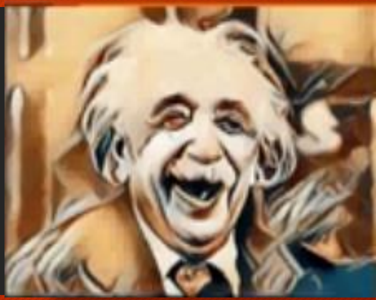
Here is a BiBTeX citation as well:

@misc{Lichman:2013 , author = "M. Lichman",

year = "2013", title = "{UCI} Machine Learning Repository",

url = "http://archive.ics.uci.edu/ml", institution = "University of California, Irvine, School of Information and Computer Sciences" }

Hands On – R Machine Learning Ex-14

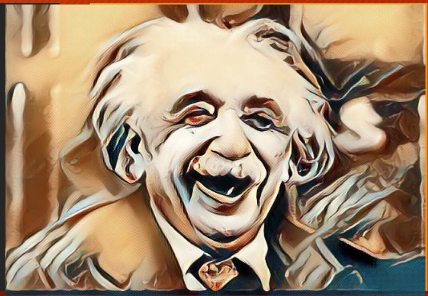


- Use the same dataset as used for k-NN algorithm, but using C5.0 predict the values in the following dataset.
- http://archive.ics.uci.edu/ml/datasets/Glass+Identification

- Using the Balance Scale Data Set, predict using Rpart tree, please refer the regression Rpart model for hints.
- http://archive.ics.uci.edu/ml/datasets/Balance+Scale

- Compare k-NN, C5.0 and Rpart models.

Videos



- ✓ C5.0 Decision Tree Intro - DIY- 25 -of-50

  https://www.youtube.com/watch?v=VS4eMbpc43w

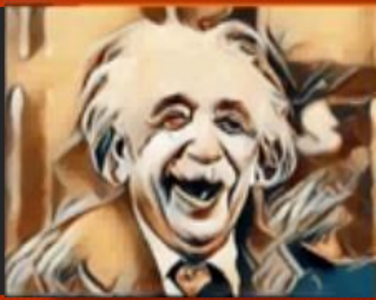
- ✓ C5.0 Decision Tree Use Case - DIY- 26 -of-50

  https://www.youtube.com/watch?v=mpDM4fYPKFM

- ✓ C5.0 Decision Tree Exercise - DIY- 27 -of-50

  https://www.youtube.com/watch?v=EFNcuw5-1Vc

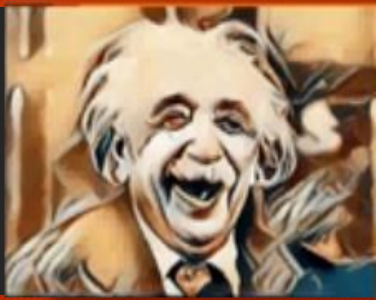
Random Forest Tree - Classification



- Random Forest – also known as Decision trees forest are very powerful classifiers, which utilize a collection of tree structures to model the relationships among the features and the potential outcomes.
- The basic idea of a Random Forest is to select random features to add diversity to the decision trees. After the forest/a collection of trees is generated, the model uses a vote to combine the trees' predictions. An all-purpose classifier, which has an ability to select only the most important features.
- RandomForest syntax:

```
randomForest_model<- randomForest(train_Predictors, train_Target, ntree=100)
randomForest_predict<- predict(randomForest_model, test_data, type=<response or vote or prob> )
```

Random Forest Tree - Classification



- Get the data from Balance Scale Data Set.
- Attribute Information:
  - Class Name: 3 (L, B, R)
  - Left-Weight: 5 (1, 2, 3, 4, 5)
  - Left-Distance: 5 (1, 2, 3, 4, 5)
  - Right-Weight: 5 (1, 2, 3, 4, 5)
  - Right-Distance: 5 (1, 2, 3, 4, 5)
- http://archive.ics.uci.edu/ml/datasets/Balance+Scale

Citation Policy:

If you publish material based on databases obtained from this repository, then, in your acknowledgements, please note the assistance you received by using this repository. This will help others to obtain the same data sets and replicate your experiments. We suggest the following pseudo-APA reference format for referring to this repository:

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

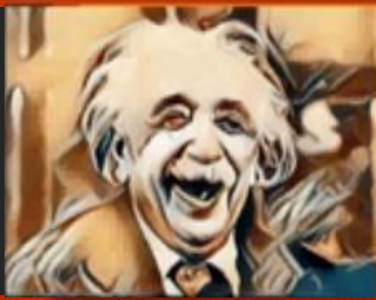
Here is a BiBTeX citation as well:

@misc{Lichman:2013 , author = "M. Lichman",

year = "2013", title = "{UCI} Machine Learning Repository",

url = "http://archive.ics.uci.edu/ml", institution = "University of California, Irvine, School of Information and Computer Sciences" }

Hands On – R Machine Learning Ex-15



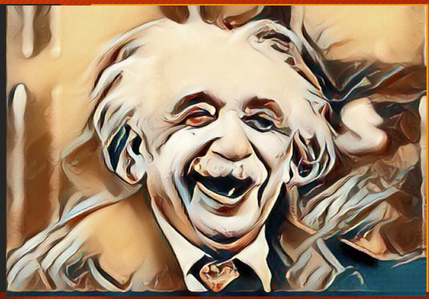
- Use the same dataset as used for k-NN & C5.0 algorithms, but using randomForest predict the values in the following dataset.
- http://archive.ics.uci.edu/ml/datasets/Glass+Identification

- Using the Balance Scale Data Set, predict using k-NN & C5.0 trees.
- http://archive.ics.uci.edu/ml/datasets/Balance+Scale

- Using the different "types" in the prediction step in randomForest.

- Compare k-NN, C5.0 and randomForest models for the above datasets.

Videos



- ✓ Random Forest Intro - DIY- 28 -of-50

  https://www.youtube.com/watch?v=2zaMHZXuMEY

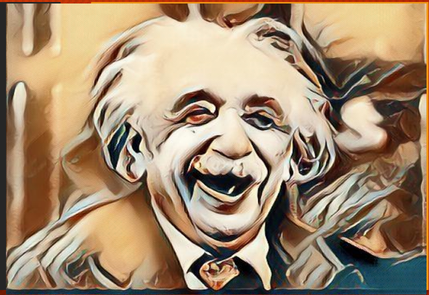
- ✓ Random Forest Hands on - DIY- 29 -of-50

  https://www.youtube.com/watch?v=1gaK5XAjxB4

- ✓ Random Forest Exercise - DIY- 30 -of-50

  https://www.youtube.com/watch?v=VrhDd4becPg

Naïve Bayes – Probabilistic Classification

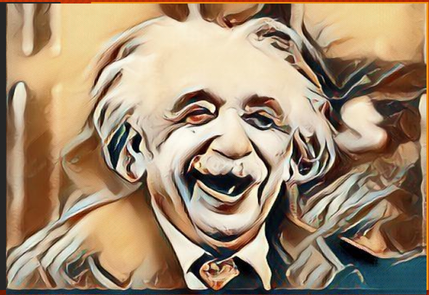


- ✓ Naïve Bayes technique descended from the work of a mathematician Thomas Bayes, who developed foundational principles to describe the probability of events, and how probabilities they change with additional information.

- ✓ This method utilizes training data to calculate an observed probability of each outcome based on the feature values.
  - ✓ p(A|B) = (p(A) * p(B|A)) / p(B)

- ✓ This trained classifier when applied to an unlabeled data, predicts the most likely class for the new features using the observed probabilities.

Naïve Bayes – Probabilistic Classification

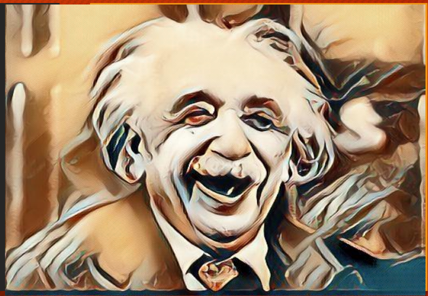


- ✓ It assumes that all of the features in the dataset are equally important and independent. These assumptions aren’t always correct.

- ✓ Naïve Bayes model still preforms well, when these assumptions are violated, as it’s very versatile and accurate across many types of conditions it is often a strong first candidate for classification learning tasks.

- ✓ Naïve Bayes algorithm works best on categorical data as it uses frequency table to learn the number of occurrences. However, this feature doesn’t work best on Numeric data; a solution to this problem is to create bins.

Naïve Bayes – Probabilistic Classification



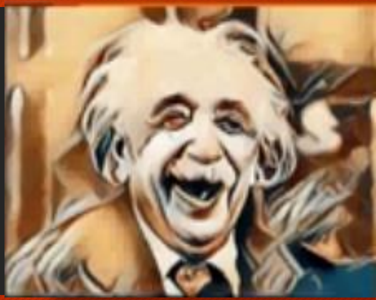
- ✓ Get the data from UCI YouTube+Spam+Collection

  Dataset Citation Request:
  We would appreciate:

  1. If you find this collection useful, make a reference to the paper below and the web page: [Web Link].
  2. Send us a message either to talmeida < AT > ufscar.br or tuliocasagrande < AT > acm.org in case you make use of the corpus.

  http://dcomp.sor.ufscar.br/talmeida/youtubespamcollection/
- ✓ Load and clean up the data, divide the text into individual words using tm_map().
  - ✓ #lower case
  - ✓ #remove stopwords / filler words such as to, and, but etc.
  - ✓ #remove punctuations
  - ✓ #remove numbers
  - ✓ #strip white spaces;
  - ✓ #Stemming;
- ✓ Create a DTM sparse matrix – a table with the frequency of words in each line.
- ✓ Naïve Bayes Model – without & with Laplace Estimator

Hands On – R Machine Learning Ex-16



- Use the same datasets as downloaded from UCI– one at a time, and implement the Naïve Bayes to predict the values in the following dataset.
- http://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection

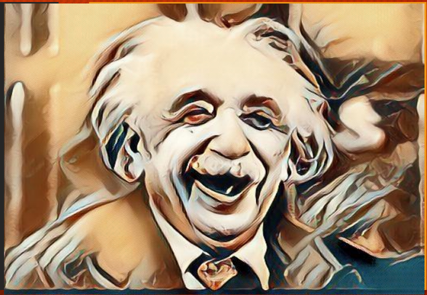
- **Dataset Citation Request:**

  We would appreciate:

  1. If you find this collection useful, make a reference to the paper below and the web page: [Web Link].
  2. Send us a message either to talmeida < AT > ufscar.br or tuliocasagrande < AT > acm.org in case you make use of the corpus.

  - http://dcomp.sor.ufscar.br/talmeida/youtubespamcollection/

Videos



- ✓ Naive Bayes - DIY- 31 -of-50

  https://www.youtube.com/watch?v=p2sxlkir804

- ✓ Naive Bayes Handson- DIY- 32 -of-50

  https://www.youtube.com/watch?v=JYBnamUVOEU

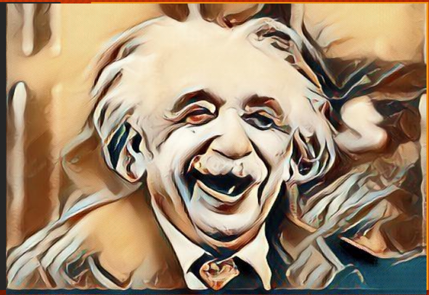
- ✓ Naive Bayes Handson contd- DIY- 33 -of-50

  https://www.youtube.com/watch?v=w_bLm5BeYvo

- ✓ Naive Bayes Exercise- DIY- 34 -of-50

  https://www.youtube.com/watch?v=aaOyUT6baIY

Apriori Algorithm – Association Rule

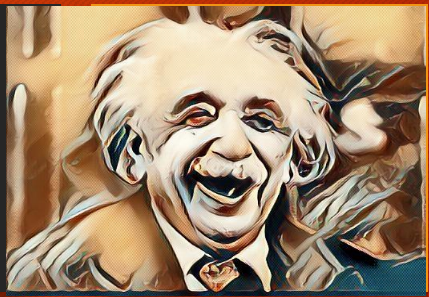


- ✓ Association analysis uses a set of transactions to discover rules that indicate the likely occurrence of an item based on the occurrences of other items in the transaction.

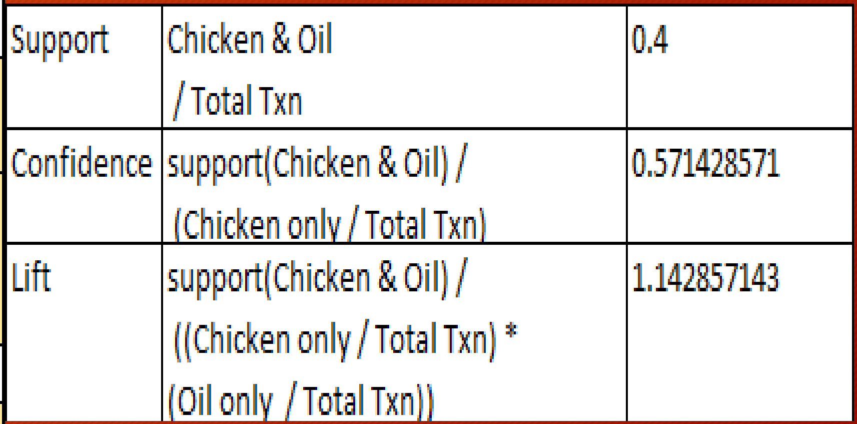
- ✓ Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers in retail stores, or details of a website visit frequency).

- ✓ Common usage:
  - ✓ Market Basket Analysis: Products bought together frequently.
  - ✓ In the field of healthcare for the detection of adverse drug reactions. It produces association rules that indicates what all combinations of medications and patient characteristics lead to ADRs.

Apriori Algorithm – Example

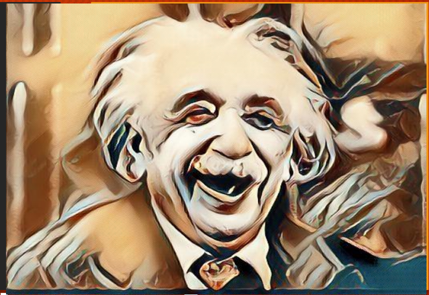


Transaction	Products	Chicken Only	Chicken & Oil	Oil Only
1	Chicken, Oil, Onion, Milk	1	1	1
2	Milk, Bread, Potato			
3	Chicken, Bread, Oil	1	1	1
4	Bread, Oil, Onion, Chicken	1	1	1
5	Potato, Salt, Milk			
6	Oil, Milk, Bread, Potato			1
7	Chicken, Salt	1		
8	Onion, Milk, Potato, Chicken	1		
9	Potato, Chicken	1		
10	Milk, Oil, Yogurt, Chicken	1	1	1
		7	**4**	**5**



Support	Chicken & Oil / Total Txn	0.4
Confidence	support(Chicken & Oil) / (Chicken only / Total Txn)	0.571428571
Lift	support(Chicken & Oil) / ((Chicken only / Total Txn) * (Oil only / Total Txn))	1.142857143

Apriori Algorithm – Association Rule

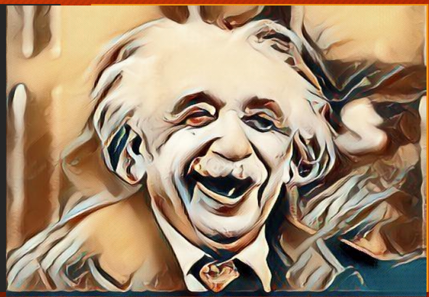


- ✓ Support - the rule is denoted as sup(X➔ Y) and is the number of transactions where XUY appears divided by the total number of transactions.

- ✓ Confidence - the confidence is the number of transactions where XUY appears divided by the number of transactions where only X appears.

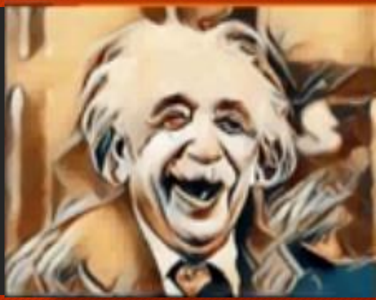
- ✓ Lift - A lift value greater than 1 indicates that X and Y appear more often together than expected; this means that the occurrence of X has a positive effect on the occurrence of Y or that X is positively correlated with Y.

Apriori Algorithm – Association Rule



- ✓ You would want all three to be high.
  - ✓ high support: should apply to a large amount of cases
  - ✓ high confidence: should be correct often
  - ✓ high lift: indicates it is not just a coincidence

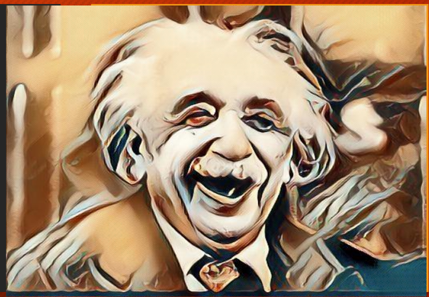
Hands On – R Machine Learning Ex-17



- Get the Titanic: Machine Learning from Disaster data set from the following link, and predict survival on the Titanic passengers.

- https://www.kaggle.com/c/titanic/data

Videos



- ✓ Apriori Algorithm Concepts- DIY- 35 -of-50

  https://www.youtube.com/watch?v=Ds2O_0xw4kc

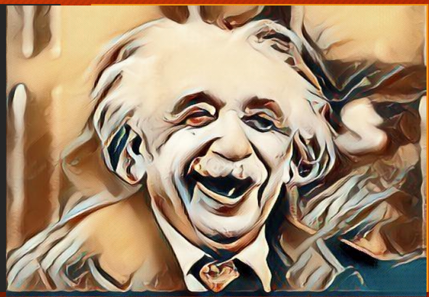
- ✓ Support Confidence Lift - Apriori- DIY- 36 -of-50

  https://www.youtube.com/watch?v=ztmgtLggSFU

- ✓ Apriori Hands-on Example - DIY- 37 -of-50

  https://www.youtube.com/watch?v=NW48UkpZH8E

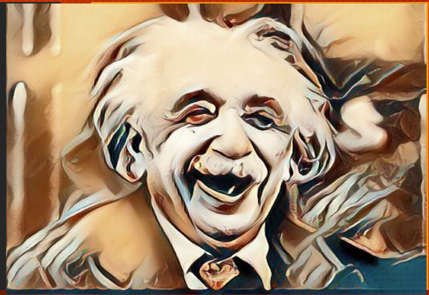
Forecasting – Time Series Models



- ✓ ARIMA - Auto-Regressive(p) Integrated(d) Moving Average(q)- time series forecasting models are very popular, adaptive, flexible models, which utilize historical information to make predictions of a value into the future.

- ✓ Time series analysis can not only be used in various business applications for forecasting a quantity into the future, and explaining its historical patterns; examples:
  - ❖ Estimating the effect of a newly launched product line or a product.
  - ❖ Predicting the price value of stocks.
  - ❖ Forecasting and predicting seasonal patterns in a product / services sales.

Forecasting – Time Series Models

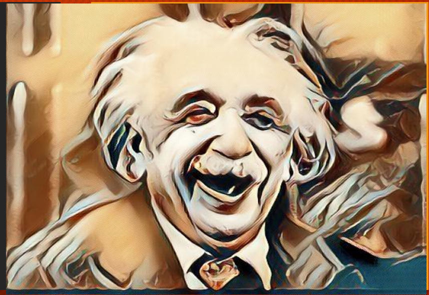


- ✓ ARIMA - Auto-Regressive(p) Integrated(d) Moving Average(q)

- ✓ The forecast package allows you to specify the order of the model using the arima() function.
  - ❖ E.g: arima(1,0,1)

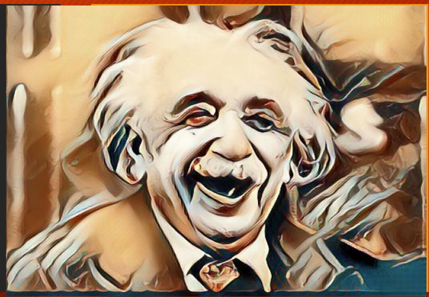
- ✓ One can choose to automatically generate a set of optimal (p, d, q) values using *auto.arima()*, which searches through various combinations of order parameters, and selects the optimal values. *auto.arima()* also allows the user to specify maximum order for (p, d, q), which is set to 5 by default.

Forecasting – Time Series Models



- ✓ The building blocks of a time series analysis are seasonality, trend, and cycle.
- ✓ Seasonality refers to fluctuations in the data related to calendar cycles.
- ✓ Cycle components refers the non-seasonal patterns in the data.
- ✓ Trend refers to the overall patterns in the data series.
- ✓ Residual or Error is a part of the time series that can't be attributed to seasonal, cycle, or trend components.

Forecasting – Time Series Models - STL

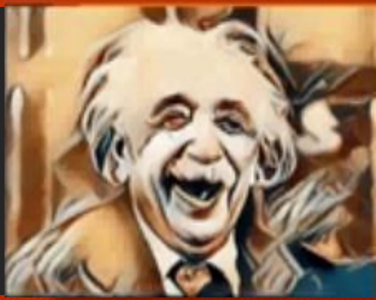


- ✓ STL is a function for decomposing and forecasting the series. Once can calculate seasonal component of the data using stl() using smoothing. It by default assumes additive model structure.

- ✓ Since ARIMA uses previous lags of series to model its behavior, modeling stable series with consistent properties involves less uncertainty.

- ✓ Fitting an ARIMA model requires the series to be stationary meaning its mean, variance, and auto-covariance are time invariant.

Hands On – R Machine Learning Ex-18



- Use the Hour.csv Bike Sharing Dataset to create an Arima model.
- Arima model should also have – Auto ARIMA, Seasonal as well as non-seasonal.

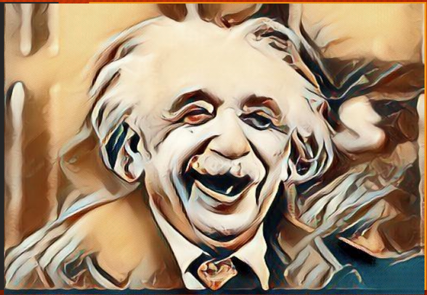
https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset

- **Citation Request:**

Fanaee-T, Hadi, and Gama, Joao, 'Event labeling combining ensemble detectors and background knowledge', Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg, [Web Link].

```
@article{
year={2013},
issn={2192-6352},
journal={Progress in Artificial Intelligence},
doi={10.1007/s13748-013-0040-3},
title={Event labeling combining ensemble detectors and background knowledge},
url={[Web Link]},
publisher={Springer Berlin Heidelberg},
keywords={Event labeling; Event detection; Ensemble learning; Background knowledge},
author={Fanaee-T, Hadi and Gama, Joao},
pages={1-15}
}
```

Videos



- ✓ ARIMA Time Series - DIY- 38 -of-50

  https://www.youtube.com/watch?v=joeqXDb2nXE

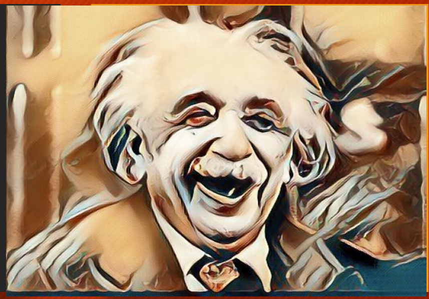
- ✓ ARIMA Hands On - DIY- 39 -of-50

  https://www.youtube.com/watch?v=VQ7fcV6LJHY

- ✓ ARIMA Seasonality - DIY- 40 -of-50

  https://www.youtube.com/watch?v=HoObvOkLF2k

R - Command Line



- ✓ Run Rscript.exe from <Rinstall_Location>\bin

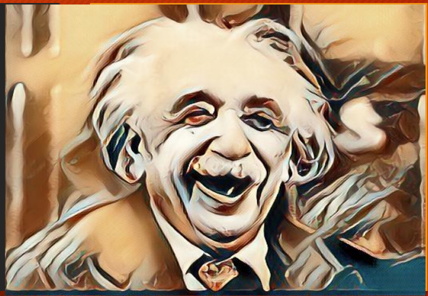
  c:\Program Files\R\R-3.2.5\bin>

- ✓ To run a simple R script

  Rscript.exe <FileLocation>\<FileName>.R

- ✓ Passing argument to R script
  - Add args = commandArgs(trailingOnly=TRUE); in the R Script.
  - *Rscript.exe <FileLocation>\<FileName>.R <Arg1> <Arg2>*
    Rscript.exe bdcs.R PassArg1 PassArg2
  - *args[1] will be PassArg1*
  - *args[2] will be PassArg2*

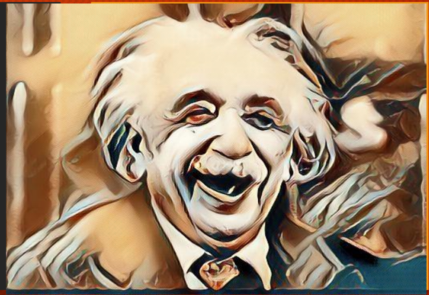
Videos



- ✓ R Script Command Line - DIY- 41 -of-50

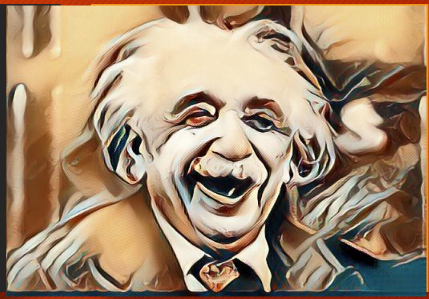
  https://www.youtube.com/watch?v=rOjUvm-AvUI

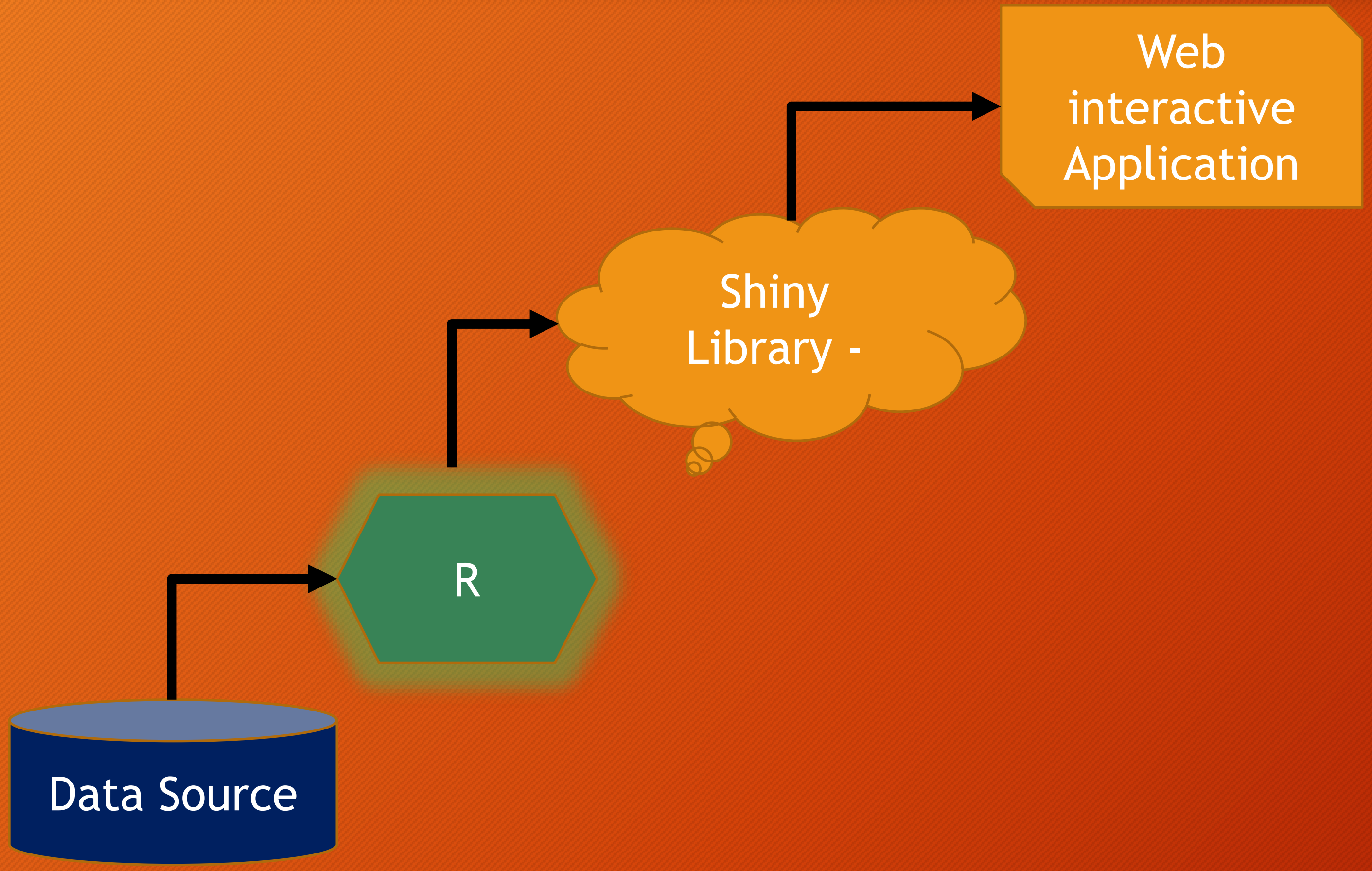
Deploying you R application with Shiny

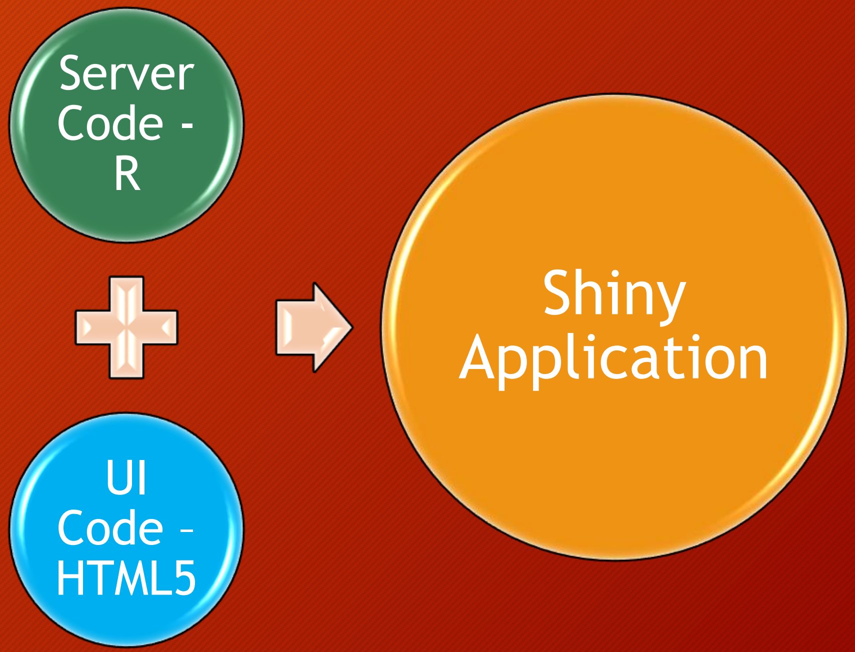


- ✓ **Shiny** is an open source **R** package, which combines the computational power of R with the interactivity of the modern web.
- ✓ It provides a powerful web framework for building web applications using **R**.
- ✓ **Shiny** helps you turn your analyses into interactive web applications without requiring HTML, CSS, or JavaScript knowledge.
- ✓ Enables standalone apps on a webpage or embed them in R Markdown documents or build dashboards.

Shiny & R Architecture







Shiny Code Components

- ✓ To use **Shiny,** initiate library.
- ✓ There are two components of the code – ui and server.
- ✓ You can create a single app.R with ui and server codes or you separately in ui.R and Server.R

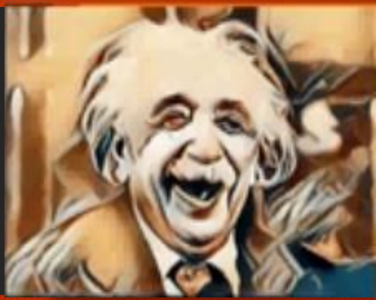
```
library(shiny)
ui <- fluidPage()
server <- function(input, output) {}
shinyApp(ui = ui, server = server)
```

Shiny Code Components

- ✓ To use **Shiny,** initiate library.
- ✓ There are two components of the code – ui and server.
- ✓ You can create a single app.R with ui and server codes or you separately in ui.R and Server.R

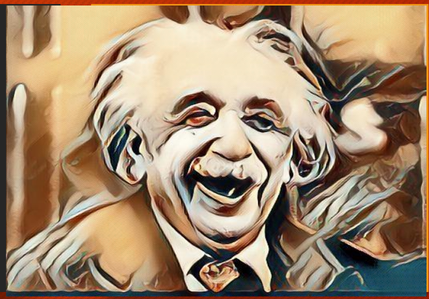
```
library(shiny)
ui <- fluidPage(‘This is Machine Learning Do it yourself’)
server <- function(input, output) {}
shinyApp(ui = ui, server = server)
```

Hands On – R Machine Learning Ex-19



- Create a basic structure of a Shiny app.
- Goto https://shiny.rstudio.com/ and click on getting started – read more about Shiny apps.

Videos



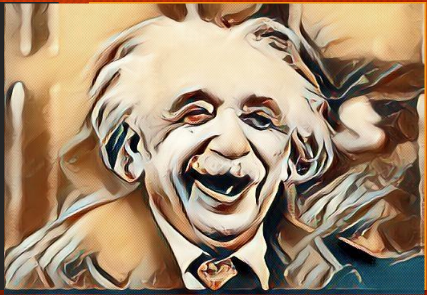
- ✓ Introduction to Shiny Apps - DIY- 42 -of-50

  https://www.youtube.com/watch?v=IgHHXcSfM7c

- ✓ Creating a Shiny App - DIY- 43 -of-50

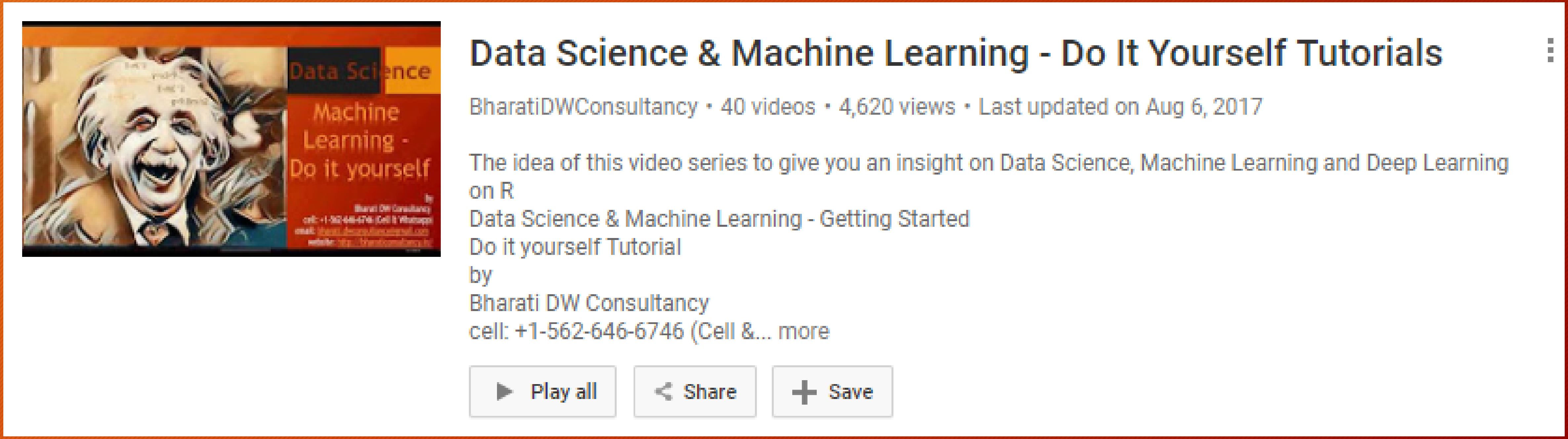
  https://www.youtube.com/watch?v=dwVbzAIRAEc

Videos Series – Complete Playlist



- ✓ Data Science & Machine Learning - Do It Yourself Tutorials

https://www.youtube.com/watch?v=sc4WxfJFvh4&list=PLyD1XCIRA3gTfulK7bZnqSkPLeSVffKpQ

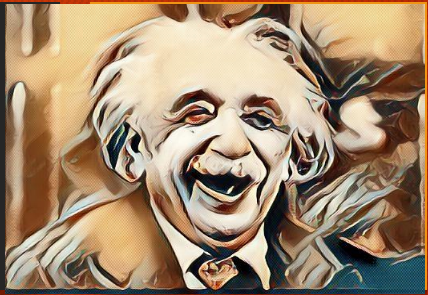


Data Science & Machine Learning - Do It Yourself Tutorials

BharatiDWConsultancy • 40 videos • 4,620 views • Last updated on Aug 6, 2017

The idea of this video series to give you an insight on Data Science, Machine Learning and Deep Learning on R
Data Science & Machine Learning - Getting Started
Do it yourself Tutorial
by
Bharati DW Consultancy
cell: +1-562-646-6746 (Cell &... more

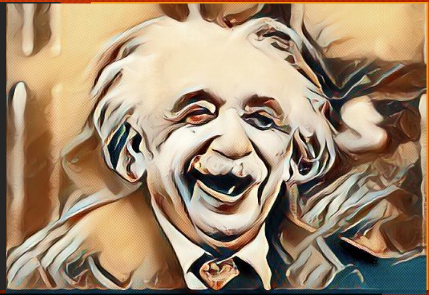
Videos Series – Complete Playlist - I



Content	URL
Getting Started - DIY- 1 -of-50	https://www.youtube.com/watch?v=sc4WxfJFvh4
R Data Structures - DIY- 2 -of-50	https://www.youtube.com/watch?v=Ht8O5JW9Grl
R Data Structures - Factors - DIY- 3 -of-50	https://www.youtube.com/watch?v=6tgUbEBJnJA
R Data Structures - List & Matrices - DIY- 4 -of-50	https://www.youtube.com/watch?v=GApMimnormc
R Data Structures - Data Frames - DIY- 5 -of-50	https://www.youtube.com/watch?v=8SEOtWRZ91k
Frequently used R commands - DIY- 6 -of-50	https://www.youtube.com/watch?v=FaWpgQVYMMU
Frequently used R commands contd - DIY- 7 -of-50	https://www.youtube.com/watch?v=ES-1_zLYcRE
Installing RStudio- DIY- 8 -of-50	https://www.youtube.com/watch?v=82Uo73d_JXc
R Data Visualization Basics - DIY- 9 -of-50	https://www.youtube.com/watch?v=72alsyrW3dU
Linear Regression Model - DIY- 10(a) -of-50	https://www.youtube.com/watch?v=zQkRSKcanIU

Content	URL
Linear Regression Model - DIY- 10(b) -of-50	https://www.youtube.com/watch?v=3KVco80HRnE
Multiple Linear Regression Model - DIY- 11 -of-50	https://www.youtube.com/watch?v=orD_H8avzOg
Evaluate Model Performance - DIY- 12 -of-50	https://www.youtube.com/watch?v=uMeQc_crllc
RMSE & R-Squared - DIY- 13 -of-50	https://www.youtube.com/watch?v=pYFqFUGObts
Numeric Predictions using Regression Trees - DIY- 14 -of-50	https://www.youtube.com/watch?v=Np1wXBdizvl
Regression Decision Trees contd - DIY- 15 -of-50	https://www.youtube.com/watch?v=mt0Dv_2izgo
Method Types in Regression Trees - DIY- 16 -of-50	https://www.youtube.com/watch?v=qO9_RQgnSGA
Real Time Project 1 - DIY- 17 -of-50	https://www.youtube.com/watch?v=AzCN-BFu-cE
Helping to create solution to Real Time Project#1 - DIY- 18 -of-50	https://www.youtube.com/watch?v=yMQJMf9uEZs
KNN Classification - DIY- 21 -of-50	https://www.youtube.com/watch?v=AC0j_rtFZRg

Videos Series – Complete Playlist - II



Content	URL
KNN Classification Hands on - DIY- 22 -of-50	https://www.youtube.com/watch?v=S9Hv3KRrnyU
KNN Classification HandsOn Contd - DIY- 23 -of-50	https://www.youtube.com/watch?v=S94y56-iLws
KNN Classification Exercise - DIY- 24 -of-50	https://www.youtube.com/watch?v=s_h4EzqO_Eo
C5.0 Decision Tree Intro - DIY- 25 -of-50	https://www.youtube.com/watch?v=VS4eMbpc43w
C5.0 Decision Tree Use Case - DIY- 26 -of-50	https://www.youtube.com/watch?v=mpDM4fYPKFM
C5.0 Decision Tree Exercise - DIY- 27 -of-50	https://www.youtube.com/watch?v=EFNcuw5-1Vc
Random Forest Intro - DIY- 28 -of-50	https://www.youtube.com/watch?v=2zaMHZXuMEY
Random Forest Hands on - DIY- 29 -of-50	https://www.youtube.com/watch?v=1gaK5XAjxB4
Random Forest Exercise - DIY- 30 -of-50	https://www.youtube.com/watch?v=VrhDd4becPg
Naive Bayes - DIY- 31 -of-50	https://www.youtube.com/watch?v=p2sxlkir804
Naive Bayes Handson- DIY- 32 -of-50	https://www.youtube.com/watch?v=JYBnamUVOEU
Naive Bayes Handson contd- DIY- 33 -of-50	https://www.youtube.com/watch?v=w_bLm5BeYvo
Naive Bayes Exercise- DIY- 34 -of-50	https://www.youtube.com/watch?v=aaOyUT6baIY
Apriori Algorithm Concepts- DIY- 35 -of-50	https://www.youtube.com/watch?v=Ds2O_0xw4kc
Support Confidence Lift - Apriori- DIY- 36 -of-50	https://www.youtube.com/watch?v=ztmgtLggSFU
Apriori Hands-on Example - DIY- 37 -of-50	https://www.youtube.com/watch?v=NW48UkpZH8E
ARIMA Time Series - DIY- 38 -of-50	https://www.youtube.com/watch?v=joeqXDb2nXE
ARIMA Hands On - DIY- 39 -of-50	https://www.youtube.com/watch?v=VQ7fcV6LJHY
ARIMA Seasonality - DIY- 40 -of-50	https://www.youtube.com/watch?v=HoObvOkLF2k
R Script Command Line - DIY- 41 -of-50	https://www.youtube.com/watch?v=rOjUvm-AvUI
Introduction to Shiny Application	https://www.youtube.com/watch?v=lgHHXcSfM7c
Creating a Shiny Application.	https://www.youtube.com/watch?v=dwVbzAIRAEc