Zeppelin

DS-670_Assignment...

default

1. Data Preparation

FINISHED

```
# Here I have calculated returns from price and normalization of selected indicator data is done.
```

Took 0 sec. Last updated by anonymous at March 25 2017, 8:38:20 AM. (outdated)

FINISHED

```
%spark.r

small_cap_stock <- read.csv("/home/scarface/Desktop/sem-3/capstone/madhu/stockPerform/iwm.csv", header=TRU
large_cap_stock <- read.csv("/home/scarface/Desktop/sem-3/capstone/madhu/stockPerform/spy.csv", header=TRU

View(small_cap_stock)
View(large_cap_stock)
typeof(large_cap_stock$Date)
```

```
[1] “integer”
```

Took 1 sec. Last updated by anonymous at March 25 2017, 4:42:24 AM. (outdated)

FINISHED

```
%spark.r
large_cap_stock2000 <- subset(large_cap_stock, large_cap_stock$Date >= "2017-02-07")
large_cap_stock2000
```

```
[1] Date      Open      High      Low       Close     Volume    Adj.Close
&lt;0 rows&gt; (or 0-length row.names)
```

Took 0 sec. Last updated by anonymous at March 25 2017, 4:43:26 AM. (outdated)

FINISHED

```
%spark.r
arqdata <- read.csv(file = "/home/scarface/Desktop/sem-3/capstone/madhu/ARQ_Data.csv", header=TRUE)
save(arqdata, file="arq.rdata")
load("arq.rdata")
nrow(arqdata)
```

```
[1] 89206
```

Took 20 sec. Last updated by anonymous at March 25 2017, 4:44:42 AM. (outdated)

FINISHED

```
%spark.r
View(arqdata)
t(names(arqdata))
arq[72]
#Considering the column price with data
arqdata_na <- subset(arqdata, arqdata$price != "NA")
View(arqdata_na)
```

```
[1,] “ticker” “dimension” “calendardate” “datekey” “reportperiod” “accoci”
     [,7]     [,8]        [,9]      [,10]      [,11]           [,12]
[1,] “assets” “assetsavg” “assetsc” “assetsnc” “assetturnover” “bvps”
     [,13]   [,14]     [,15]        [,16] [,17]          [,18] [,19]
[1,] “capex” “cashneq” “cashnequsd” “cor” “currentratio” “de”  “debt”
     [,20]     [,21]     [,22]      [,23] [,24]  [,25]    [,26]<br />
[1,] “debtusd” “depamor” “divyield” “dps” “ebit” “ebitda” “ebitdamargin”
```

```
     [,27]     [,28]     [,29] [,30] [,31]  [,32]    [,33]<br />
[1,] "ebitdausd" "ebitusd" "ebt" "eps" "epsdil" "epsusd" "equity"
     [,34]     [,35]     [,36] [,37]  [,38]     [,39] [,40]<br />
[1,] "equityavg" "equityusd" "ev"  "evebit" "evebitda" "fcf" "fcfps"
     [,41]  [,42] [,43]    [,44]    [,45]    [,46]<br />
[1,] "fxusd" "gp"  "grossmargin" "intangibles" "intexp" "invcap"
     [,47]      [,48]      [,49]         [,50]          [,51]<br />
[1,] "invcapavg" "inventory" "liabilities" "liabilitiesc" "liabilitiesnc"
     [,52]      [,53] [,54]      [,55]     [,56]    [,57]  [,58]  [,59]
[1,] "marketcap" "ncf" "ncfcommon" "ncfdebt" "ncfdiv" "ncff" "ncfi" "ncfo"
     [,60]  [,61]    [,62]       [,63]          [,64]       [,65]<br />
[1,] "ncfx" "netinc" "netinccmn" "netinccmnusd" "netincdis" "netmargin"
     [,66]      [,67]         [,68] [,69] [,70] [,71]       [,72]   [,73]
[1,] "payables" "payoutratio" "pb"  "pe"  "pe1" "prefdivis" "price" "ps"
     [,74] [,75]          [,76]     [,77]     [,78]        [,79] [,80]
[1,] "ps1" "receivables" "retearn" "revenue" "revenueusd" "rnd" "roa"
     [,81] [,82]  [,83] [,84]  [,85]         [,86]        [,87]<br />
[1,] "roe" "roic" "ros" "sgna" "sharefactor" "sharesbas" "shareswa"
     [,88]         [,89] [,90]       [,91]    [,92]   [,93]<br />
[1,] "shareswadil" "sps" "tangibles" "taxexp" "tbvps" "workingcapital"
```

Took 1 min 16 sec. Last updated by anonymous at March 25 2017, 4:47:16 AM. (outdated)

FINISHED

```
%spark.r
##################################Data_File - with ARQ listings########################################
return <- vector();
for (i in 2:length(arqdata_na[,1]))
{
  if (identical(arqdata_na[i,1],arqdata_na[i-1,1]))
  {
    #    return[i] = ((arqdata[i,72] / arqdata[i-1,72]) - 1);
    return[i] = (arqdata_na[i,72] / arqdata_na[i-1,72]);
  }
  else
  {
    return[i] = 0;
  }
}

return[1]=0;

```

Took 18 sec. Last updated by anonymous at March 25 2017, 4:53:46 AM. (outdated)

FINISHED

```
%spark.r
#adding return to arqdata dataset
arqdata_returns <- cbind(arqdata_na,return)
View(arqdata_returns)
t(names(arqdata_returns))
View(arqdata_returns)
```

```
[1,] "ticker" "dimension" "calendardate" "datekey" "reportperiod" "accoci"
     [,7]     [,8]        [,9]      [,10]      [,11]           [,12]
[1,] "assets" "assetsavg" "assetsc" "assetsnc" "assetturnover" "bvps"
     [,13]   [,14]     [,15]         [,16] [,17]           [,18] [,19]
[1,] "capex" "cashneq" "cashnequsd" "cor" "currentratio" "de"  "debt"
     [,20]     [,21]     [,22]       [,23] [,24]  [,25]    [,26]<br />
[1,] "debtusd" "depamor" "divyield" "dps" "ebit" "ebitda" "ebitdamargin"
     [,27]     [,28]     [,29] [,30] [,31]    [,32]    [,33]<br />
[1,] "ebitdausd" "ebitusd" "ebt" "eps" "epsdil" "epsusd" "equity"
     [,34]     [,35]       [,36] [,37]    [,38]      [,39] [,40]<br />
[1,] "equityavg" "equityusd" "ev"  "evebit" "evebitda" "fcf" "fcfps"
```

```
     [,41]    [,42] [,43]          [,44]          [,45]     [,46]<br />
[1,] "fxusd" "gp"  "grossmargin" "intangibles" "intexp" "invcap"
     [,47]       [,48]       [,49]          [,50]          [,51]<br />
[1,] "invcapavg" "inventory" "liabilities" "liabilitiesc" "liabilitiesnc"
     [,52]       [,53] [,54]        [,55]     [,56]    [,57]  [,58]  [,59]
[1,] "marketcap" "ncf" "ncfcommon" "ncfdebt" "ncfdiv" "ncff" "ncfi" "ncfo"
```


Took 1 min 15 sec. Last updated by anonymous at March 25 2017, 4:55:27 AM. (outdated)

FINISHED

```
%spark.r
#Indicators listed.
#Calculating ratios by choice 1.SGNA/REVENUE (sgnamargin) and 2.ebitmargin
sgnamargin = arqdata_returns$sgna / arqdata_returns$revenue
ebitmargin = arqdata_returns$ebit / arqdata_returns$revenue

#adding ratios by choice to dataset - arqdata_returns
arqdata_returns_ratios <- cbind(arqdata_returns, sgnamargin,ebitmargin)
View(arqdata_returns_ratios)
t(names(arqdata_returns_ratios))

```

```
[1,] "ticker" "dimension" "calendardate" "datekey" "reportperiod" "accoci"
     [,7]     [,8]         [,9]      [,10]      [,11]           [,12]
[1,] "assets" "assetsavg" "assetsc" "assetsnc" "assetturnover" "bvps"
     [,13]   [,14]     [,15]         [,16] [,17]          [,18] [,19]
[1,] "capex" "cashneq" "cashnequsd" "cor" "currentratio" "de"  "debt"
     [,20]     [,21]     [,22]      [,23] [,24]  [,25]    [,26]<br />
[1,] "debtusd" "depamor" "divyield" "dps" "ebit" "ebitda" "ebitdamargin"
     [,27]       [,28]     [,29] [,30] [,31]    [,32]    [,33]<br />
[1,] "ebitdausd" "ebitusd" "ebt" "eps" "epsdil" "epsusd" "equity"
     [,34]       [,35]       [,36] [,37]    [,38]      [,39] [,40]<br />
[1,] "equityavg" "equityusd" "ev"  "evebit" "evebitda" "fcf" "fcfps"
     [,41]   [,42] [,43]          [,44]          [,45]    [,46]<br />
[1,] "fxusd" "gp"  "grossmargin" "intangibles" "intexp" "invcap"
     [,47]       [,48]       [,49]          [,50]          [,51]<br />
[1,] "invcapavg" "inventory" "liabilities" "liabilitiesc" "liabilitiesnc"
     [,52]       [,53] [,54]       [,55]     [,56]    [,57]  [,58]  [,59]
[1,] "marketcap" "ncf" "ncfcommon" "ncfdebt" "ncfdiv" "ncff" "ncfi" "ncfo"
     [,60]  [,61]    [,62]        [,63]            [,64]        [,65]<br />
```

Took 42 sec. Last updated by anonymous at March 25 2017, 4:56:57 AM. (outdated)

FINISHED

```
%spark.r
#Consider the 20 indicators chosen
#factors required in the dataset
arq_data_factors <- arqdata_returns_ratios[c(1,3,77,16,42,84,24,43,95,96,45,91,61,29,30,65,7,9,50,17,93,13
head(arq_data_factors)

#17 lines
```

```
ticker calendardate   revenue      cor       gp     sgna     ebit
1      A   31-03-2011 1.519e+09 7.03e+08 8.16e+08 4.46e+08 2.21e+08
2      A   30-06-2011 1.677e+09 7.77e+08 9.00e+08 4.69e+08 2.80e+08
3      A   30-09-2011 1.691e+09 7.99e+08 8.92e+08 4.49e+08 3.01e+08
4      A   31-12-2011 1.728e+09 8.07e+08 9.21e+08 4.45e+08 3.16e+08
5      A   31-03-2012 1.635e+09 7.61e+08 8.74e+08 4.41e+08 2.82e+08
6      A   30-06-2012 1.733e+09 8.15e+08 9.18e+08 4.52e+08 3.18e+08
  grossmargin sgnamargin ebitmargin  intexp   taxexp   netinc      ebt
1       0.537  0.2936142  0.1454905 2.3e+07  5.0e+06 1.93e+08 1.98e+08
2       0.537  0.2796661  0.1669648 2.0e+07  6.0e+07 2.00e+08 2.60e+08
3       0.527  0.2655234  0.1780012 2.0e+07 -4.9e+07 3.30e+08 2.81e+08
4       0.533  0.2575231  0.1828704 2.3e+07  4.0e+06 2.89e+08 2.93e+08
```

DS-670_Assignment9_StockAnalysis

```
3   0.535  0.2697248  0.1724771 2.6e+07  2.6e+07 2.30e+08 2.56e+08
4   0.[illegible] 0.2[illegible] 0.1834968 2.5e+07  3.8e+07 2.55e+08 2.93e+08
   eps netmargin    assets   assetsc liabilitiesc currentratio
1 0.56     0.127 8.044e+09 4.598e+09    1.406e+09        3.270
2 0.56     0.119 8.549e+09 5.096e+09    1.592e+09        3.201
```

Took 0 sec. Last updated by anonymous at March 25 2017, 8:47:08 AM.

FINISHED

```
%spark.r
caldate = unique(arq_data_factors$calendardate)
length(caldate)
prj2_arq_date = vector();
prj2_arq_nn = vector();
prj2_date_replace = vector();
factors1 <- NULL
factors2 <- NULL
factors3 <- NULL
#35 lines
```

```
[1] 20
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:00:03 AM. (outdated)

FINISHED

```
%spark.r
# Loop for dates - each date we get a dataset prj2_arq_'date'
for (i in 1:length(caldate)){
  #if (i < 16)
  #{
    #print(paste0("calendar date: ", caldate[i]))
    factors1 <- subset(arq_data_factors, calendardate == caldate[i])

    ### Calculating log(returns) ###
    factors1 <- subset(factors1, factors1$return != 0)
    return_log <- log(factors1$return)
    factors1 <- cbind(factors1, return_log)

    ### Remove all NAs in our dataset before normalizing
    factors1 <- na.omit(factors1)

    ### Normalizing all Indicators
    revenue_nor <- (factors1[,3] - mean(factors1[,3])) / sd(factors1[,3])
    cor_nor <- (factors1[,4] - mean(factors1[,4])) / sd(factors1[,4])
    gp_nor <- (factors1[,5] - mean(factors1[,5])) / sd(factors1[,5])
    sgna_nor <- (factors1[,6] - mean(factors1[,6])) / sd(factors1[,6])
    ebit_nor <- (factors1[,7] - mean(factors1[,7])) / sd(factors1[,7])
    gm_nor <- (factors1[,8] - mean(factors1[,8])) / sd(factors1[,8])
    sgna_mg_nor <- (factors1[,9] - mean(factors1[,9])) / sd(factors1[,9])
    ebit_mg_nor <- (factors1[,10] - mean(factors1[,10])) / sd(factors1[,10])
    intexp_nor <- (factors1[,11] - mean(factors1[,11])) / sd(factors1[,11])
    taxexp_nor <- (factors1[,12] - mean(factors1[,12])) / sd(factors1[,12])
    netinc_nor <- (factors1[,13] - mean(factors1[,13])) / sd(factors1[,13])
    ebt_nor <- (factors1[,14] - mean(factors1[,14])) / sd(factors1[,14])
    eps_nor <- (factors1[,15] - mean(factors1[,15])) / sd(factors1[,15])
    netmargin_nor <- (factors1[,16] - mean(factors1[,16])) / sd(factors1[,16])
    assets_nor <- (factors1[,17] - mean(factors1[,17])) / sd(factors1[,17])
    assetsc_nor <- (factors1[,18] - mean(factors1[,18])) / sd(factors1[,18])
    liabc_nor <- (factors1[,19] - mean(factors1[,19])) / sd(factors1[,19])
    cur_ratio_nor <- (factors1[,20] - mean(factors1[,20])) / sd(factors1[,20])
    wc_nor <- (factors1[,21] - mean(factors1[,21])) / sd(factors1[,21])
    capex_nor <- (factors1[,22] - mean(factors1[,22])) / sd(factors1[,22])

    ### Appending normalized columns to new factors2

    factors1 <- cbind(factors1, revenue_nor,cor_nor,gp_nor,sgna_nor,ebit_nor,gm_nor,sgna_mg_nor,ebit_mg_n
                      intexp_nor,taxexp_nor,netinc_nor,ebt_nor,eps_nor,netmargin_nor,assets_nor,assetsc_no
                      liabc_nor,cur_ratio_nor,wc_nor,capex_nor)
    factors2 <- factors1[c(1,2,25:44,23,24)]
    factors3 <- factors1[c(25:44,24)]

    prj2_date_replace[i] <- gsub("-", "_", caldate[i])
    prj2_arq_date[i] <- paste("prj2_arq_", prj2_date_replace[i], sep = "")
    assign(prj2_arq_date[i], factors2)

    prj2_date_replace[i] <- gsub("-", "_", caldate[i])
```

```
    prj2_arq_nn[i] <- paste("prj2_arqnn_", prj2_date_replace[i], sep = "")
    assign(prj2_arq_nn[i], factors3)

    factors1 <- NULL
    factors2 <- NULL
    factors3 <- NULL

  #}
}
#70 lines
```

Took 1 sec. Last updated by anonymous at March 25 2017, 5:04:36 AM. (outdated)

FINISHED

```
%spark.r
# Datasets names
head(prj2_arq_date)
head(prj2_arq_nn)

#head(get(prj2_arq_nn[1])
#get(prj2_arq_nn[1])

```

[1] “prj2_arq_31_03_2011” “prj2_arq_30_06_2011” “prj2_arq_30_09_2011”
[4] “prj2_arq_31_12_2011” “prj2_arq_31_03_2012” “prj2_arq_30_06_2012”
[1] “prj2_arqnn_31_03_2011” “prj2_arqnn_30_06_2011” “prj2_arqnn_30_09_2011”
[4] “prj2_arqnn_31_12_2011” “prj2_arqnn_31_03_2012” “prj2_arqnn_30_06_2012”

Took 0 sec. Last updated by anonymous at March 25 2017, 9:04:54 AM.

2. Neural Network modeling

FINISHED

```
# For every quarter neural network model was run and weights from each model were gathered in a matrix for furth
```

Took 36 sec. Last updated by anonymous at March 26 2017, 8:02:42 PM. (outdated)

FINISHED

```
%spark.r
install.packages("neuralnet", repos = "http://cran.us.r-project.org")
install.packages("MASS", repos = "http://cran.us.r-project.org")

```

The downloaded source packages are in
'/tmp/RtmpdfiJRz/downloaded_packages'

Took 19 sec. Last updated by anonymous at March 25 2017, 5:06:39 AM. (outdated)

FINISHED

```
%spark.r
library("MASS")
library("neuralnet")
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:07:01 AM. (outdated)

FINISHED

```
%spark.r
###################### neural network - package does not take Strings!!
names_date <- names(get(prj2_arq_nn[1]))
names_date
```

[1] “revenue_nor” “cor_nor” “gp_nor” “sgna_nor”
[5] “ebit_nor” “gm_nor” “sgna_mg_nor” “ebit_mg_nor”
[9] “intexp_nor” “taxexp_nor” “netinc_nor” “ebt_nor”
[13] “eps_nor” “netmargin_nor” “assets_nor” “assetsc_nor”
[17] “liabc_nor” “cur_ratio_nor” “wc_nor” “capex_nor”

```
[21] "return_log"
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:07:07 AM. (outdated)

FINISHED default

```
%spark.r
### log(returns) as y ... we need to get the formula to use in nueral networks as it does not take strin
names_date %in% "return_log"
!names_date %in% "return_log"
paste(names_date[!names_date %in% "return_log"])
paste(names_date[!names_date %in% "return_log"], collapse = "+")
paste("return_log ~ ", paste(names_date[!names_date %in% "return_log"], collapse = "+"))
formula1 <- as.formula(paste("return_log ~ ", paste(names_date[!names_date %in% "return_log"], collapse =
formula1

```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
 [1] “revenue_nor”   “cor_nor”       “gp_nor”        “sgna_nor”<br />
 [5] “ebit_nor”      “gm_nor”        “sgna_mg_nor”   “ebit_mg_nor”<br />
 [9] “intexp_nor”    “taxexp_nor”    “netinc_nor”    “ebt_nor”<br />
[13] “eps_nor”       “netmargin_nor” “assets_nor”    “assetsc_nor”<br />
[17] “liabc_nor”     “cur_ratio_nor” “wc_nor”        “capex_nor”<br />
[1] “revenue_nor+cor_nor+gp_nor+sgna_nor+ebit_nor+gm_nor+sgna_mg_nor+ebit_mg_nor+intexp_nor+taxexp_nor+netinc_no
r+ebt_nor+eps_nor+netmargin_nor+assets_nor+assetsc_nor+liabc_nor+cur_ratio_nor+wc_nor+capex_nor”
[1] “return_log ~  revenue_nor+cor_nor+gp_nor+sgna_nor+ebit_nor+gm_nor+sgna_mg_nor+ebit_mg_nor+intexp_nor+taxexp
_nor+netinc_nor+ebt_nor+eps_nor+netmargin_nor+assets_nor+assetsc_nor+liabc_nor+cur_ratio_nor+wc_nor+capex_nor”
return_log ~ revenue_nor + cor_nor + gp_nor + sgna_nor + ebit_nor +
    gm_nor + sgna_mg_nor + ebit_mg_nor + intexp_nor + taxexp_nor +
    netinc_nor + ebt_nor + eps_nor + netmargin_nor + assets_nor +
    assetsc_nor + liabc_nor + cur_ratio_nor + wc_nor + capex_nor
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:07:11 AM. (outdated)

FINISHED

```
%spark.r
############# Assigning datasets - from date 15 to date 20
uvw1 <- get(prj2_arq_nn[15])
uvw2 <- get(prj2_arq_nn[16])
uvw3 <- get(prj2_arq_nn[17])
uvw4 <- get(prj2_arq_nn[18])
uvw5 <- get(prj2_arq_nn[19])
uvw6 <- get(prj2_arq_nn[20])
head(get(prj2_arq_date[17]))
#94 lines
```

```
17       A   31-03-2015  0.01439033 -0.04207989  0.1390798  0.20840226
38      AA   31-03-2015  1.20665857  1.26814971  0.7906741  0.09887883
44     AAC   31-03-2015 -0.23017689 -0.21310986 -0.2159198 -0.19253635
83     AAL   31-03-2015  2.20365650  1.19547031  3.9814477  3.57694991
146   AAOI   31-03-2015 -0.23330843 -0.20638101 -0.2406638 -0.21766917
167   AAON   31-03-2015 -0.22173300 -0.19478331 -0.2317944 -0.21520489
        ebit_nor      gm_nor sgna_mg_nor ebit_mg_nor   intexp_nor
17   0.005610863  0.05788391 -0.04398179  0.04982287 -0.005237661
38   0.743086384 -0.07349110 -0.04627883  0.04981099  1.976632243
44  -0.150845233  0.30670023 -0.04162526  0.04988165 -0.290533443
83   1.756284735  0.09271820 -0.04421367  0.04990082  3.621958201
146 -0.159739786 -0.02571837 -0.04472714  0.04939392 -0.302050725
167 -0.136494358 -0.04960474 -0.04567916  0.05011704 -0.304387835
     taxexp_nor netinc_nor      ebt_nor    eps_nor netmargin_nor
17  -0.07104492  0.0357906  0.006291527 0.01802950    0.05025438
38   1.22135355  0.3202280  0.587601802 0.01791388    0.05011986
```

Zeppelin

Took 1 sec. Last updated by anonymous at March 25 2017, 9:05:34 AM.

DS-670_Assignment...

FINISHED default

```
%spark.r

############# Assigning datasets - from date 1 to date 15
mar2011 <- get(prj2_arq_nn[1])
jun2011 <- get(prj2_arq_nn[2])
sep2011 <- get(prj2_arq_nn[3])
dec2011 <- get(prj2_arq_nn[4])
mar2012 <- get(prj2_arq_nn[5])
jun2012 <- get(prj2_arq_nn[6])
sep2012 <- get(prj2_arq_nn[7])
dec2012 <- get(prj2_arq_nn[8])
mar2013 <- get(prj2_arq_nn[9])
jun2013 <- get(prj2_arq_nn[10])
sep2013 <- get(prj2_arq_nn[11])
dec2013 <- get(prj2_arq_nn[12])
mar2014 <- get(prj2_arq_nn[13])
jun2014 <- get(prj2_arq_nn[14])
sep2014 <- get(prj2_arq_nn[15])
dec2014 <- get(prj2_arq_nn[16])
mar2015 <- get(prj2_arq_nn[17])
jun2015 <- get(prj2_arq_nn[18])
sep2015 <- get(prj2_arq_nn[19])
dec2015 <- get(prj2_arq_nn[20])

#mar2011


```

Took 1 sec. Last updated by anonymous at March 25 2017, 9:06:00 AM.

FINISHED

```
%spark.r
###################### neural network - package does not
names_date <- names(get(prj2_arq_nn[1]))
names_date
### log(returns) as y .... we need to get the formula to use in nueral networks as it does not take strin
names_date %in% "return_log"
!names_date %in% "return_log"
paste(names_date[!names_date %in% "return_log"])
paste(names_date[!names_date %in% "return_log"], collapse = "+")
paste("return_log ~ ", paste(names_date[!names_date %in% "return_log"], collapse = "+"))
formula1 <- as.formula(paste("return_log ~ ", paste(names_date[!names_date %in% "return_log"], collapse =
formula1
#124 lines
```

```
[1] “revenue_nor”   “cor_nor”       “gp_nor”        “sgna_nor”<br />
 [5] “ebit_nor”      “gm_nor”        “sgna_mg_nor”   “ebit_mg_nor”<br />
 [9] “intexp_nor”    “taxexp_nor”    “netinc_nor”    “ebt_nor”<br />
[13] “eps_nor”       “netmargin_nor” “assets_nor”    “assetsc_nor”<br />
[17] “liabc_nor”     “cur_ratio_nor” “wc_nor”        “capex_nor”<br />
[21] “return_log”<br />
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
 [1] “revenue_nor”   “cor_nor”       “gp_nor”        “sgna_nor”<br />
 [5] “ebit_nor”      “gm_nor”        “sgna_mg_nor”   “ebit_mg_nor”<br />
 [9] “intexp_nor”    “taxexp_nor”    “netinc_nor”    “ebt_nor”<br />
[13] “eps_nor”       “netmargin_nor” “assets_nor”    “assetsc_nor”<br />
[17] “liabc_nor”     “cur_ratio_nor” “wc_nor”        “capex_nor”<br />
[1] “revenue_nor+cor_nor+gp_nor+sgna_nor+ebit_nor+gm_nor+sgna_mg_nor+ebit_mg_nor+intexp_nor+taxexp_nor+netinc_
nor+ebt_nor+eps_nor+netmargin_nor+assets_nor+assetsc_nor+liabc_nor+cur_ratio_nor+wc_nor+capex_nor”
[1] “return_log ~ revenue_nor+cor_nor+gp_nor+sgna_nor+ebit_nor+gm_nor+sgna_mg_nor+ebit_mg_nor+intexp_nor+taxe
```

Took 3 sec. Last updated by anonymous at March 25 2017, 5:24:52 AM. (outdated)

DS-670_Assignment9_StockAnalysis 1. Data Preparation Untitled Untitled Untitle

Zeppelin

DS-670_Assignment...

FINISHED

default

```
%spark.r

## 2011 ##
nn1_2011_03_31 <- neuralnet(formula1, data=mar2011, hidden = c(8,7), linear.output = T)
nn1_2011_06_30 <- neuralnet(formula1, data=jun2011, hidden = c(8,7), linear.output = T, threshold = 0.05)
nn1_2011_09_30 <- neuralnet(formula1, data=sep2011, hidden = c(8,7), linear.output = T, threshold = 0.05)
nn1_2011_12_31 <- neuralnet(formula1, data=dec2011, hidden = c(8,7), linear.output = T, threshold = 0.05)
## weights 2011 ##
wgt_2011_03_31 <- nn1_2011_03_31$result.matrix
wgt_2011_06_30 <- nn1_2011_06_30$result.matrix
wgt_2011_09_30 <- nn1_2011_09_30$result.matrix
wgt_2011_12_31 <- nn1_2011_12_31$result.matrix

## 2012 ##
nn1_2012_03_31 <- neuralnet(formula1, data=mar2012, hidden = c(8,7), linear.output = T, threshold = 0.05)
nn1_2012_06_30 <- neuralnet(formula1, data=jun2012, hidden = c(8,7), linear.output = T, threshold = 0.05)
nn1_2012_09_30 <- neuralnet(formula1, data=sep2012, hidden = c(8,7), linear.output = T, threshold = 0.05)
nn1_2012_12_31 <- neuralnet(formula1, data=dec2012, hidden = c(8,7), linear.output = T, threshold = 0.06)
## weights 2012 ##
wgt_2012_03_31 <- nn1_2012_03_31$result.matrix
wgt_2012_06_30 <- nn1_2012_06_30$result.matrix
wgt_2012_09_30 <- nn1_2012_09_30$result.matrix
wgt_2012_12_31 <- nn1_2012_12_31$result.matrix

## 2013 ##
nn1_2013_03_31 <- neuralnet(formula1, data=mar2013, hidden = c(8,7), linear.output = T, threshold = 0.05)
nn1_2013_06_30 <- neuralnet(formula1, data=jun2013, hidden = c(8,7), linear.output = T, threshold = 0.05)
nn1_2013_09_30 <- neuralnet(formula1, data=sep2013, hidden = c(8,7), linear.output = T, threshold = 0.05)
nn1_2013_12_31 <- neuralnet(formula1, data=dec2013, hidden = c(8,7), linear.output = T, threshold = 0.07)
## weights 2013 ##
wgt_2013_03_31 <- nn1_2013_03_31$result.matrix
wgt_2013_06_30 <- nn1_2013_06_30$result.matrix
wgt_2013_09_30 <- nn1_2013_09_30$result.matrix
wgt_2013_12_31 <- nn1_2013_12_31$result.matrix

## 2014 ##
nn1_2014_03_31 <- neuralnet(formula1, data=mar2014, hidden = c(8,7), linear.output = T, threshold = 0.05)
nn1_2014_06_30 <- neuralnet(formula1, data=jun2014, hidden = c(8,7), linear.output = T, threshold = 0.05)
nn1_2014_09_30 <- neuralnet(formula1, data=uvw1, hidden = c(8,7), linear.output = T, stepmax = 1e6)
nn1_2014_12_31 <- neuralnet(formula1, data=uvw2, hidden = c(8,7), linear.output = T, stepmax = 1e6)
## weights 2014 ##
wgt_2014_03_31 <- nn1_2014_03_31$result.matrix
wgt_2014_06_30 <- nn1_2014_06_30$result.matrix
wgt_2014_09_30 <- nn1_2014_09_30$result.matrix
wgt_2014_12_31 <- nn1_2014_12_31$result.matrix

weights_matrix <- as.data.frame(cbind(wgt_2011_03_31, wgt_2011_06_30, wgt_2011_09_30, wgt_2011_12_31,
                                      wgt_2012_03_31, wgt_2012_06_30, wgt_2012_09_30, wgt_2012_12_31,
                                      wgt_2013_03_31, wgt_2013_06_30, wgt_2013_09_30, wgt_2013_12_31,
                                      wgt_2014_03_31, wgt_2014_06_30, wgt_2014_09_30))

colnames(weights_matrix) <- c("nn1_2011_03_31", "nn1_2011_06_30", "nn1_2011_09_30", "nn1_2011_12_31",
                              "nn1_2012_03_31", "nn1_2012_06_30", "nn1_2012_09_30", "nn1_2012_12_31",
                              "nn1_2013_03_31", "nn1_2013_06_30", "nn1_2013_09_30", "nn1_2013_12_31",
                              "nn1_2014_03_31", "nn1_2014_06_30", "nn1_2014_09_30")

matrix_weight <- read.csv(file = "/home/scarface/Desktop/sem-3/capstone/madhu/Matrix_Weight.csv", header=
save(matrix_weight, file="matWeights.rdata")
load("matWeights.rdata")
nrow(matrix_weight)
ncol(matrix_weight)
#164 lines
```

```
[1] 15
[1] 239
```

Took 0 sec. Last updated by anonymous at March 25 2017, 7:58:56 AM. (outdated)

3. Time Series Analysis

FINISHED

```
# Time series modeling for gathered weights is done using the Arima function to predict the weights of future 5
```

Took 0 sec. Last updated by anonymous at March 26 2017, 8:03:11 PM. (outdated)

DS-670_Assignment9_StockAnalysis

DS-670_Assignment9_StockAnalysis 1. Data Preparation Untitled Untitled Untitle

Zeppelin

DS-670_Assignment...

FINISHED

default

```
%spark.r
######## Smoothing the weights from NN model
####### Time series Analysis - MA model timeseries_nn.R
#######
################## Use p=0, d=2, q=1 as model ########################
p_w <- vector();
predict_weights = vector();
abc=vector();
for(j in 1:5)
{
  print(paste0("Date: ", j))
  for(i in 1:ncol(matrix_weight))
  {
    test1 <- arima(matrix_weight[,i], order = c(0,2,1))
    test2 <- predict(test1, n.ahead = 1)
    p_w = test2$pred[1]
    predict_weights[i] <- p_w
  }
  predict_weight <- t(predict_weights)
  #print(paste0(nrow(matrix_weight),":",nrow))
  matrix_weight = rbind(matrix_weight, predict_weight)
  #print(rbind(matrix_weight, predict_weight))
  print (paste0("Row count: ", nrow(matrix_weight)))

  p_w = NULL;
  predict_weights = NULL;
  predict_weight = NULL;
  #test1 <- NULL;
  #test2 <- NULL;
}

#View(matrix_weight)

#177 lines
```

```
[1] “Date: 1”
[1] “Row count: 16”
[1] “Date: 2”
[1] “Row count: 17”
[1] “Date: 3”
[1] “Row count: 18”
[1] “Date: 4”
[1] “Row count: 19”
[1] “Date: 5”
[1] “Row count: 20”
```

Took 3 sec. Last updated by anonymous at March 25 2017, 7:59:57 AM. (outdated)

FINISHED

```
%spark.r
nrow(matrix_weight)
ncol(matrix_weight)
#head(matrix_weight)
```

```
[1] 20
[1] 239
```

Took 0 sec. Last updated by anonymous at March 25 2017, 9:07:08 AM.

FINISHED

```
%spark.r
###################### Neural Network Equation ######################
#View(dec2014)
for (a in 16:20)
{
  if (a==16){
    for(b in 1:ncol(get(prj2_arq_nn[a]))-1)
    {
      if(b == 1)
      {
```

```
        #print(paste0("Column no:", b, get(prj2_arq_nn[a])[b]))
      }
    }
  }

}
```

Zeppelin

DS-670_Assignment...

default

Took 0 sec. Last updated by anonymous at March 25 2017, 9:07:44 AM

READY