Machine Learning in Econometrics - Group Work

Hollosi, Pokasz, Soos, Szabo

2023-12-12

About dataset

A novel dataset for bankruptcy prediction related to American public companies listed on the New York Stock Exchange and NASDAQ is provided. The dataset comprises accounting data from 8,262 distinct companies recorded during the period spanning from 1999 to 2018.

For further information: kaggle.com

Status_label column contains the flag whether the company has gone to bankrupt after the last reported year. It's permanently 'failed' for seased companies not just for the last period!

```
download.file(url = "https://raw.githubusercontent.com/hollipista/MachLearnInEcon/main/american_bankrupt
              destfile = "american_bankruptcy.zip", mode = "wb")
unzip("american_bankruptcy.zip")

df <- read_csv("american_bankruptcy.csv") %>%
  arrange(company_name, year) %>%
  group_by(company_name) %>%
  mutate(last = ifelse(row_number() == max(row_number()), 1, 0)) %>% #find last reported year for each
  ungroup()
```

```
## Rows: 78682 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): company_name, status_label
## dbl (19): year, X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
colnames <- c("X1", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9",
              "X10", "X11", "X12", "X13", "X14", "X15", "X16", "X17", "X18")

for (col in colnames) { #moving avg for each variable
  new_col_name <- paste0(col, "_rollmean")
  df <- df %>%
    group_by(company_name) %>%
    mutate(!!new_col_name := rollmean(!!sym(col), k = 3, align = "right", fill = NA)) %>%
    ungroup()
}

df <- df %>%  #flag the last year of bankrupted companies = year before bankrupcy
```

```
  mutate(bankrupt = ifelse(last == 1 & status_label == 'failed', 1, 0))

table(df$bankrupt) # number of bankrupcies
```

```
##
##     0     1
## 78073   609
```

```
# here I calculate the change of current year on last 3 years rolling avg
for (col in colnames) { #change variables
  new_col_name <- paste0(col, "_chg")
  roll_col_name <- paste0(col, "_rollmean")
  df <- df %>%
    group_by(company_name) %>%
    mutate(!!new_col_name := (!!sym(col)-lag(!!sym(roll_col_name), n=1))/abs(lag(!!sym(roll_col_name), n
    ungroup()
}

df <- df %>% # I keep years that has 4 year lead: 3 for the moving average + 1 for the change
  drop_na()

table(df$bankrupt) # number of bankrupcies
```

```
##
##     0     1
## 39219   425
```

```
colnameschg <- c("X1_chg", "X2_chg", "X3_chg", "X4_chg", "X5_chg", "X6_chg", "X7_chg", "X8_chg", "X9_chg
              "X10_chg", "X11_chg", "X12_chg", "X13_chg", "X14_chg", "X15_chg", "X16_chg", "X17_chg", "X
describe(df)
```

```
## Warning in w * sort(x - mean(x)): longer object length is not a multiple of
## shorter object length

## Warning in w * sort(x - mean(x)): longer object length is not a multiple of
## shorter object length

## Warning in w * sort(x - mean(x)): longer object length is not a multiple of
## shorter object length

## Warning in w * sort(x - mean(x)): longer object length is not a multiple of
## shorter object length

## Warning in spikecomp(x, method = "grid", lumptails = lumptails, normalize =
## FALSE, : possible logic error 1 in spikecomp

## Warning in spikecomp(x, method = "grid", lumptails = lumptails, normalize =
## FALSE, : program logic error 2 in spikecomp

## Warning in xrange[freq != 0] <- xrnz: number of items to replace is not a
## multiple of replacement length
```

```
## Warning in w * sort(x - mean(x)): longer object length is not a multiple of
## shorter object length


## df
##
##  59  Variables      39644  Observations
## ---------------------------------------------------------------------------
## company_name
##        n  missing distinct
##    39644        0     4884
##
## lowest : C_1    C_10   C_100  C_1000 C_1001, highest: C_990  C_995  C_997  C_998  C_999
## ---------------------------------------------------------------------------
## status_label
##        n  missing distinct
##    39644        0        2
##
## Value       alive failed
## Frequency   37085   2559
## Proportion  0.935  0.065
## ---------------------------------------------------------------------------
## year
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0       17    0.996     2009    5.604     2002     2003
##      .25      .50      .75      .90      .95
##     2005     2009     2013     2016     2017
##
## Value       2002  2003  2004  2005  2006  2007  2008  2009  2010  2011  2012
## Frequency   2786  2803  2770  2682  2584  2485  2422  2407  2343  2231  2151
## Proportion 0.070 0.071 0.070 0.068 0.065 0.063 0.061 0.061 0.059 0.056 0.054
##
## Value       2013  2014  2015  2016  2017  2018
## Frequency   2100  2074  2031  1981  1954  1840
## Proportion 0.053 0.052 0.051 0.050 0.049 0.046
##
## For the frequency table, variable is rounded to the nearest 0
## ---------------------------------------------------------------------------
## X1
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    36786        1     1338     2226    2.476    6.864
##      .25      .50      .75      .90      .95
##   31.479  186.716  801.013 2706.975 5393.700
##
## lowest : -0.011 0.001  0.002  0.003  0.004 , highest: 131339 135676 139660 159851 169662
## ---------------------------------------------------------------------------
## X2
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    37158        1     2523     4341    2.249    6.508
##      .25      .50      .75      .90      .95
##   35.146  248.487 1268.985 4781.661 9344.400
##
## lowest : -1.477 -0.666 0      0.001  0.002 , highest: 351176 351530 355913 362867 374623
## ---------------------------------------------------------------------------
## X3
```

```
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    24478        1    182.4    315.1   0.1582   0.4210
##      .25      .50      .75      .90      .95
##   2.2258  16.6840  86.1670 334.8400 783.2898
##
## lowest : 0     0.001 0.002 0.003 0.004, highest: 22016 22308 24387 25847 28430
## ------------------------------------------------------------------------------
## X4
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    34402        1    594.4     1064   -15.81    -5.50
##      .25      .50      .75      .90      .95
##     1.30    41.16   277.00  1142.49  2581.94
##
## lowest : -21913  -9647   -8218.5 -7236   -4467
## highest: 69905   70744   78669   81565   81730
## ------------------------------------------------------------------------------
## X5
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    28813        1    335.9    571.5    0.008    0.391
##      .25      .50      .75      .90      .95
##    3.872   31.779  188.920  694.870 1412.873
##
## lowest : 0     0.001 0.002 0.003 0.004, highest: 44858 45141 46756 47257 62567
## ------------------------------------------------------------------------------
## X6
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    32929        1    219.8    570.1 -112.072  -38.348
##      .25      .50      .75      .90      .95
##   -4.570    6.423   88.363  433.916 1047.222
##
## lowest : -98696 -38468 -29580 -23119 -21176, highest:  45220  45687  48351  53394  59531
## ------------------------------------------------------------------------------
## X7
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    31291        1    427.5    721.8    0.272    1.100
##      .25      .50      .75      .90      .95
##    6.981   47.573  243.539  889.000 1785.940
##
## lowest : -0.006 0      0.001  0.002  0.003 , highest: 34987  35673  36450  38642  48995
## ------------------------------------------------------------------------------
## X8
##        n   missing  distinct      Info      Mean       Gmd       .05       .10
##    39644         0     39450         1      4950      8643     4.367    10.507
##      .25       .50       .75       .90       .95
##   53.498   385.925  2164.280  9006.676 19809.098
##
## lowest : 3e-04   7e-04   8e-04   9e-04   0.0011
## highest: 729439  737467  757029  790050  1073390
## ------------------------------------------------------------------------------
## X9
##        n   missing  distinct      Info      Mean       Gmd       .05       .10
##    39644         0     37952         1      3719      6310     3.209    11.227
##      .25       .50       .75       .90       .95
##   60.362   424.012  2036.576  7221.652 14744.669
```

```
##
## lowest : -1.977 -0.143 0.001  0.002  0.003 , highest: 479962 482154 483521 496785 511729
## ---------------------------------------------------------------------------
## X10
##        n   missing  distinct      Info      Mean       Gmd       .05       .10
##    39644         0     38331         1      4427      7547     5.391    13.341
##       .25       .50       .75       .90       .95
##    64.208   431.260  2267.455  8974.483 20571.760
##
## lowest : 0.002  0.003  0.004  0.005  0.007 , highest: 375319 402672 403821 444097 531864
## ---------------------------------------------------------------------------
## X11
##        n   missing  distinct      Info      Mean       Gmd       .05       .10
##    39644         0     26238     0.996      1114      1941    0.0000    0.0000
##       .25       .50       .75       .90       .95
##    0.8357   41.7805  532.0250 2458.2495 5521.6923
##
## lowest : -0.023 0      0.001  0.002  0.003 , highest: 113642 113681 118515 125972 166250
## ---------------------------------------------------------------------------
## X12
##        n   missing  distinct      Info      Mean       Gmd       .05       .10
##    39644         0     33557         1       412     778.7  -30.9149  -11.7876
##       .25       .50       .75       .90       .95
##   -0.7135   20.8945  182.8620  803.3337 1804.4354
##
## lowest : -25913  -13353  -8851   -8722.5 -8715
## highest: 59476   61344   66290   70662   71230
## ---------------------------------------------------------------------------
## X13
##        n   missing  distinct      Info      Mean       Gmd       .05       .10
##    39644         0     36417         1      1196      2049    0.2002    2.8040
##       .25       .50       .75       .90       .95
##   19.0575  135.8910  644.5165 2314.3512 4675.9799
##
## lowest : -21536   -8951    -8001    -6887    -4141.33
## highest: 127608   128432   130978   133918   137106
## ---------------------------------------------------------------------------
## X14
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    35065        1    929.4     1600    1.950    4.017
##       .25      .50      .75      .90      .95
##   15.044   85.465  433.000 1832.420 4010.800
##
## lowest : 0.012  0.026  0.027  0.032  0.037 , highest: 81590  82237  85181  100814 116866
## ---------------------------------------------------------------------------
## X15
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    38313        1    967.9     2795  -807.98  -315.91
##       .25      .50      .75      .90      .95
##   -61.22    18.33   393.67  1904.97  4634.57
##
## lowest : -102362 -101456  -99586  -97728  -95527
## highest:  385592  388933  389427  398278  402089
## ---------------------------------------------------------------------------
```

```
## X16
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     37952         1      3719      6310     3.209    11.227
##       .25       .50       .75       .90       .95
##    60.362   424.012  2036.576  7221.652 14744.669
##
## lowest : -1.977 -0.143 0.001  0.002  0.003 , highest: 479962 482154 483521 496785 511729
## ------------------------------------------------------------------------------
## X17
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     37007         1      2753      4752     2.946     6.126
##       .25       .50       .75       .90       .95
##    25.525   197.506  1309.480  5764.093 13070.575
##
## lowest : 0.012  0.053  0.059  0.07   0.072 , highest: 258578 279032 279711 302090 337980
## ------------------------------------------------------------------------------
## X18
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     38249         1      3125      5302     6.801    15.033
##       .25       .50       .75       .90       .95
##    59.901   364.500  1703.546  5991.393 11666.850
##
## lowest : 0.009  0.013  0.014  0.015  0.018 , highest: 448445 448909 452560 467603 481580
## ------------------------------------------------------------------------------
## last
##        n  missing distinct     Info      Sum     Mean      Gmd
##    39644        0        2     0.31     4632   0.1168   0.2064
##
## ------------------------------------------------------------------------------
## X1_rollmean
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39073        1     1275     2120    2.836    7.175
##      .25      .50      .75      .90      .95
##   31.181  179.474  762.372 2599.560 5189.549
##
## lowest : 0.00233333 0.00566667 0.007      0.007      0.00733333
## highest: 121797     122284     126206     141408     156391
## ------------------------------------------------------------------------------
## X2_rollmean
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39140        1     2430     4179    2.712    6.939
##      .25      .50      .75      .90      .95
##   34.544  237.416 1216.610 4573.527 9136.167
##
## lowest : -0.444333    -5.18104e-16 0            4.62593e-18  1.4456e-17
## highest: 349700       352214       352873       355191       362889
## ------------------------------------------------------------------------------
## X3_rollmean
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    34501        1    173.5    299.4   0.1873   0.4600
##      .25      .50      .75      .90      .95
##   2.2379  16.0188  81.8352 323.6516 751.9900
##
## lowest : -1.27213e-17 -4.62593e-18 1.15648e-18  2.31296e-18  4.04769e-18
```

```
## highest: 20315.3      20391.3      21895.3      24083.3      26221.3
## ---------------------------------------------------------------------------
## X4_rollmean
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    38739        1    564.8     1003  -14.646   -4.955
##      .25      .50      .75      .90      .95
##    1.379   39.458  259.280 1058.950 2430.510
##
## lowest : -5378.33 -4858.33 -3078.33 -2523.47 -2219.87
## highest: 65978.3  70485    72309.7  73861.7  73916.7
## ---------------------------------------------------------------------------
## X5_rollmean
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    35659        1    321.1    545.8    0.087    0.515
##      .25      .50      .75      .90      .95
##    4.020   30.511  180.044  662.286 1361.431
##
## lowest : -1.89478e-14 -9.4739e-15  -5.92119e-15 -4.73695e-15 -2.36848e-15
## highest: 44822.7      44933.3      45641.7      45737.3      50036.7
## ---------------------------------------------------------------------------
## X6_rollmean
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    38458        1    204.3    505.3  -95.267  -36.963
##      .25      .50      .75      .90      .95
##   -5.178    4.980   76.066  388.134  928.328
##
## lowest : -34132.7 -33663.7 -30897.7 -21931.3 -21921.6
## highest: 41776.7  43313.7  46197    49144    51189.7
## ---------------------------------------------------------------------------
## X7_rollmean
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     37421         1     409.5     691.5    0.3997    1.2434
##       .25       .50       .75       .90       .95
##    7.0074   45.4217  230.9662  848.5167 1727.4602
##
## lowest : -2.66454e-15 -2.36848e-15 -5.92119e-16 -2.96059e-16 -1.89663e-16
## highest: 32049.3      32857        35304.3      35593.7      37989
## ---------------------------------------------------------------------------
## X8_rollmean
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     39628         1      4689      8167     6.124    13.261
##       .25       .50       .75       .90       .95
##    60.014   387.851  2073.811  8456.168 18656.828
##
## lowest : 7e-04      0.00296667 0.00453333 0.0055     0.00583333
## highest: 599422     603202     664552     669547     822231
## ---------------------------------------------------------------------------
## X9_rollmean
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     39313         1      3568      6058     3.738    11.440
##       .25       .50       .75       .90       .95
##    58.522   402.617  1939.901  6906.530 14283.450
##
## lowest : 0.00433333 0.00566667 0.007      0.00766667 0.008
```

```
## highest: 475004     479247     481879     486300     496889
## ------------------------------------------------------------------------------
## X10_rollmean
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     39410         1      4193      7148     5.925    14.022
##       .25       .50       .75       .90       .95
##    63.910   410.728  2142.218  8520.741 19472.433
##
## lowest : 0.00233333 0.00566667 0.007      0.012      0.015
## highest: 344353     354243     366441     416863     459927
## ------------------------------------------------------------------------------
## X11_rollmean
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     33895         1      1033      1792 6.939e-18 3.333e-02
##       .25       .50       .75       .90       .95
## 2.056e+00 4.698e+01 5.018e+02 2.290e+03 5.118e+03
##
## lowest : -1.51582e-13 -1.32635e-13 -9.4739e-14  -9.46633e-14 -7.57912e-14
## highest: 106558       107593       108316       119389       135301
## ------------------------------------------------------------------------------
## X12_rollmean
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     38599         1     391.3       731  -27.7762  -10.3892
##       .25       .50       .75       .90       .95
##   -0.5999   20.1822  170.1962  742.3761 1708.6167
##
## lowest : -8986.33 -8612.33 -7198.67 -4893.33 -4393
## highest: 57577.3  60294.7  61069.7  63827.3  64016.7
## ------------------------------------------------------------------------------
## X13_rollmean
##         n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    38995        1     1139     1946    0.211    2.914
##      .25      .50      .75      .90      .95
##   18.422  128.206  605.190 2175.840 4440.683
##
## lowest : -4816.33 -4384.33 -2372.3  -2019.3  -1725.67
## highest: 125304   127033   129006   131109   134001
## ------------------------------------------------------------------------------
## X14_rollmean
##         n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    38566        1    883.2     1519    2.042    4.223
##      .25      .50      .75      .90      .95
##   14.875   82.195  409.769 1760.520 3866.108
##
## lowest : 0.0576667 0.059     0.0673333 0.0693333 0.0863333
## highest: 74354.7   77845.7   81436     86810     98895.3
## ------------------------------------------------------------------------------
## X15_rollmean
##         n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39366        1      909     2601  -712.28  -278.27
##      .25      .50      .75      .90      .95
##   -54.77    17.82   365.04  1756.37  4319.66
##
## lowest : -99158.3 -98124.3 -96135.7 -95871.3 -95146.7
```

```
## highest: 373226   385022   387984   390934   395320
## ---------------------------------------------------------------------------
## X16_rollmean
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     39313         1      3568      6058     3.738    11.440
##       .25       .50       .75       .90       .95
##    58.522   402.617  1939.901  6906.530 14283.450
##
## lowest : 0.00433333 0.00566667 0.007      0.00766667 0.008
## highest: 475004     479247     481879     486300     496889
## ---------------------------------------------------------------------------
## X17_rollmean
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     39093         1      2590      4469     3.149     6.234
##       .25       .50       .75       .90       .95
##    24.860   188.337  1235.358  5432.993 12396.188
##
## lowest : 0.0863333 0.111667  0.13      0.133333  0.155667
## highest: 223747    231096    254883    286944    306594
## ---------------------------------------------------------------------------
## X18_rollmean
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     39313         1      3004      5097     7.295    15.268
##       .25       .50       .75       .90       .95
##    58.198   344.547  1624.450  5744.470 11373.113
##
## lowest : 0.0153333 0.016     0.0166667 0.017     0.0263333
## highest: 440577    445824    449971    456203    467248
## ---------------------------------------------------------------------------
## bankrupt
##        n  missing distinct     Info      Sum     Mean      Gmd
##    39644        0        2    0.032      425  0.01072  0.02121
##
## ---------------------------------------------------------------------------
## X1_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39640        1   0.2576    0.712 -0.50044 -0.33811
##      .25      .50      .75      .90      .95
## -0.11455  0.08214  0.30469  0.66342  1.08807
##
## lowest : -1.01226  -0.999767 -0.999692 -0.999353 -0.998524
## highest: 166.135   206.474   302.096   440.206   782.938
## ---------------------------------------------------------------------------
## X2_chg
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     39599         1 7.661e+13 1.532e+14  -0.52470  -0.31466
##       .25       .50       .75       .90       .95
##  -0.07516   0.09536   0.29322   0.63417   1.03249
##
## lowest : -10.155     -1.00723    -1          -0.999763   -0.999574
## highest: 2606.65     4448.88     1.57217e+13 2.16173e+14 3.03707e+18
## ---------------------------------------------------------------------------
## X3_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
```

```
##    39644        0    39388        1      Inf      NaN -0.55222 -0.36051
##      .25      .50      .75      .90      .95
## -0.11022  0.08152  0.31660  0.73311  1.24538
##
## lowest : -1          -0.999588   -0.999365   -0.99921    -0.998958
## highest: 1.35108e+13 1.08086e+14 7.56605e+14 1.14139e+17 Inf
## ------------------------------------------------------------------------------
## X4_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39638        1    2.054    7.018  -1.6469  -0.8044
##      .25      .50      .75      .90      .95
##  -0.1868   0.1344   0.5074   1.3188   2.5231
##
## lowest : -3114.12 -1544    -1411.1  -910.14  -765.714
## highest: 717.864  1273.79  1670.29  1763.2   71985.3
## ------------------------------------------------------------------------------
## X5_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    37384        1      Inf      NaN -0.84604 -0.47273
##      .25      .50      .75      .90      .95
## -0.14087  0.07909  0.32825  0.84878  1.47659
##
## lowest : -1          -0.999982   -0.99936    -0.999215   -0.999172
## highest: 1.1027e+18  2.13708e+18 4.50245e+18 4.648e+18   Inf
## ------------------------------------------------------------------------------
## X6_chg
##          n    missing   distinct       Info       Mean        Gmd        .05
##      39644          0      39635          1 -5.297e+12   1.08e+13    -5.2209
##        .10        .25        .50        .75        .90        .95
##    -2.1747    -0.4766     0.1958     0.8762     2.2609     4.5947
##
## lowest : -2.12007e+17 -22790       -10676       -8307.1      -2528.5
## highest: 3263         3386.43      6015.75      7517         1.9996e+15
## ------------------------------------------------------------------------------
## X7_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39179        1      Inf      NaN -0.63614 -0.41251
##      .25      .50      .75      .90      .95
## -0.13408  0.08895  0.34693  0.81290  1.40000
##
## lowest : -1.01514    -1          -0.999704   -0.999499   -0.998921
## highest: 7.60928e+16 1.97582e+17 3.23394e+17 1.05903e+18 Inf
## ------------------------------------------------------------------------------
## X8_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39644        1   0.4022    1.193  -0.7691  -0.6101
##      .25      .50      .75      .90      .95
##  -0.2844   0.0782   0.4718   1.0922   1.7727
##
## lowest : -0.999972 -0.999967 -0.999888 -0.999854 -0.999802
## highest: 208.158   209.655   283.323   1668.25   1976.06
## ------------------------------------------------------------------------------
## X9_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
```

```
##    39644        0    39636        1   0.5054    1.146 -0.46856 -0.28254
##      .25      .50      .75      .90      .95
## -0.06221  0.10279  0.30254  0.65335  1.08336
##
## lowest : -1.10955  -1.04637  -0.999973 -0.999905 -0.999861
## highest: 673.906   912.129   1757.43   2438      2944.36
## ------------------------------------------------------------------------------
## X10_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39644        1   0.2756   0.7051 -0.46085 -0.29960
##      .25      .50      .75      .90      .95
## -0.08031  0.08866  0.28708  0.64356  1.07301
##
## lowest : -0.997859 -0.997367 -0.996974 -0.99684  -0.996706
## highest: 203.274   258.164   345.39    491.789   1161.31
## ------------------------------------------------------------------------------
## X11_chg
##         n   missing  distinct      Info      Mean       Gmd       .05       .10
##     39644         0     32380     0.998       Inf       NaN -1.000000 -1.000000
##       .25       .50       .75       .90       .95
## -0.449342  0.001146  0.544439  1.957359  7.801837
##
## -5e+18 (19616, 0.495), 0 (19338, 0.488), 5e+18 (3, 0.000), 1e+19 (2, 0.000),
## 1.5e+19 (3, 0.000), 2e+19 (1, 0.000), 2.5e+19 (2, 0.000), 8e+19 (1, 0.000),
## 1e+20 (1, 0.000), 2.1e+20 (1, 0.000), 2.25e+20 (1, 0.000), 3.7e+20 (1, 0.000),
## 6e+20 (1, 0.000), Inf (673, 0.017)
##
## For the frequency table, variable is rounded to the nearest 5e+18
## ------------------------------------------------------------------------------
## X12_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39638        1      Inf      NaN  -2.4625  -1.1654
##      .25      .50      .75      .90      .95
##  -0.2734   0.1442   0.6114   1.6557   3.2833
##
## lowest : -6329.13 -1846    -1469.33 -775.25  -764.977
## highest: 2434.67  3017.85  5919.74  9533     Inf
## ------------------------------------------------------------------------------
## X13_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39634        1   0.3609    1.584  -0.5900  -0.3456
##      .25      .50      .75      .90      .95
##  -0.0807   0.1123   0.3471   0.8291   1.4554
##
## lowest : -485.942 -478.194 -454     -324.328 -312.585
## highest: 506.649  673.027  762.761  772.143  921.449
## ------------------------------------------------------------------------------
## X14_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39641        1   0.2654   0.6972 -0.48395 -0.34143
##      .25      .50      .75      .90      .95
## -0.12098  0.09347  0.36714  0.83670  1.34246
##
## lowest : -0.99757  -0.994723 -0.992749 -0.985934 -0.979384
```

```
## highest: 38.411    48.1586   98.3233   151.454   198.897
## ---------------------------------------------------------------------------
## X15_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39644        1   0.1403    3.355 -1.98762 -0.96374
##      .25      .50      .75      .90      .95
## -0.27091  0.05024  0.35041  0.90215  1.65281
##
## lowest : -900.875 -770.962 -683.652 -597.538 -573.366
## highest: 1151.6   1422.36  1843.25  3264.32  5127
## ---------------------------------------------------------------------------
## X16_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39636        1   0.5054    1.146 -0.46856 -0.28254
##      .25      .50      .75      .90      .95
## -0.06221  0.10279  0.30254  0.65335  1.08336
##
## lowest : -1.10955  -1.04637  -0.999973 -0.999905 -0.999861
## highest: 673.906   912.129   1757.43   2438      2944.36
## ---------------------------------------------------------------------------
## X17_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39640        1   0.3235   0.7827 -0.45306 -0.29789
##      .25      .50      .75      .90      .95
## -0.09672  0.08676  0.35190  0.87759  1.45239
##
## lowest : -0.995691 -0.986839 -0.986653 -0.979634 -0.978449
## highest: 100.017   115.267   115.759   151.576   316.338
## ---------------------------------------------------------------------------
## X18_chg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    39644        0    39643        1    0.195   0.5043 -0.40342 -0.25018
##      .25      .50      .75      .90      .95
## -0.05784  0.09741  0.27923  0.56482  0.84153
##
## lowest : -0.997885 -0.995914 -0.988464 -0.986433 -0.984708
## highest: 44.8461   78.151    126.351   170.897   275.746
## ---------------------------------------------------------------------------
```

```
# There are a couple of extreme high values especially in case of X5 and X11:
# inventory and long-term debt. As both variables could be zero and the division
# by zero could result infinite floating.

df %>%
  summarise(across(where(is.numeric), ~ quantile(., probs = 0.995, na.rm = TRUE))) %>%
  t()
```

```
##                    [,1]
## year         2.018000e+03
## X1           3.401863e+04
## X2           6.587212e+04
## X3           4.617280e+03
## X4           1.624058e+04
## X5           7.935270e+03
```

```
## X6           8.185605e+03
## X7           9.843570e+03
## X8           1.459194e+05
## X9           9.136165e+04
## X10          1.109163e+05
## X11          2.523046e+04
## X12          1.227811e+04
## X13          3.254049e+04
## X14          2.629956e+04
## X15          3.655021e+04
## X16          9.136165e+04
## X17          6.465469e+04
## X18          7.950192e+04
## last         1.000000e+00
## X1_rollmean  3.118187e+04
## X2_rollmean  6.144953e+04
## X3_rollmean  4.452120e+03
## X4_rollmean  1.520801e+04
## X5_rollmean  7.546299e+03
## X6_rollmean  7.737383e+03
## X7_rollmean  9.701138e+03
## X8_rollmean  1.374022e+05
## X9_rollmean  8.861963e+04
## X10_rollmean 1.044833e+05
## X11_rollmean 2.278153e+04
## X12_rollmean 1.144604e+04
## X13_rollmean 3.166899e+04
## X14_rollmean 2.502591e+04
## X15_rollmean 3.418035e+04
## X16_rollmean 8.861963e+04
## X17_rollmean 6.185766e+04
## X18_rollmean 7.722627e+04
## bankrupt     1.000000e+00
## X1_chg       4.645873e+00
## X2_chg       6.588721e+00
## X3_chg       6.056772e+00
## X4_chg       2.228063e+01
## X5_chg                Inf
## X6_chg       5.007957e+01
## X7_chg       1.357649e+01
## X8_chg       7.199749e+00
## X9_chg       6.258183e+00
## X10_chg      4.598442e+00
## X11_chg               Inf
## X12_chg      3.458419e+01
## X13_chg      1.024127e+01
## X14_chg      5.417478e+00
## X15_chg      1.387367e+01
## X16_chg      6.258183e+00
## X17_chg      6.981353e+00
## X18_chg      3.310383e+00
```

```
# I'm winsorizing the extreme increases at +300%
df <- df %>%
```

```
  mutate(across(all_of(colnameschg), ~ pmin(3, .)))

df <- df %>%
  select(all_of(c(colnameschg, "bankrupt")))
```

Prepared dataset

1. I've calculated 3-month moving average for all variables
2. Get the yearly change for all: (t2-t1)/t1
3. Winsorized the extreme values (because some statments could be zero)
4. Dropped unnecessary variables

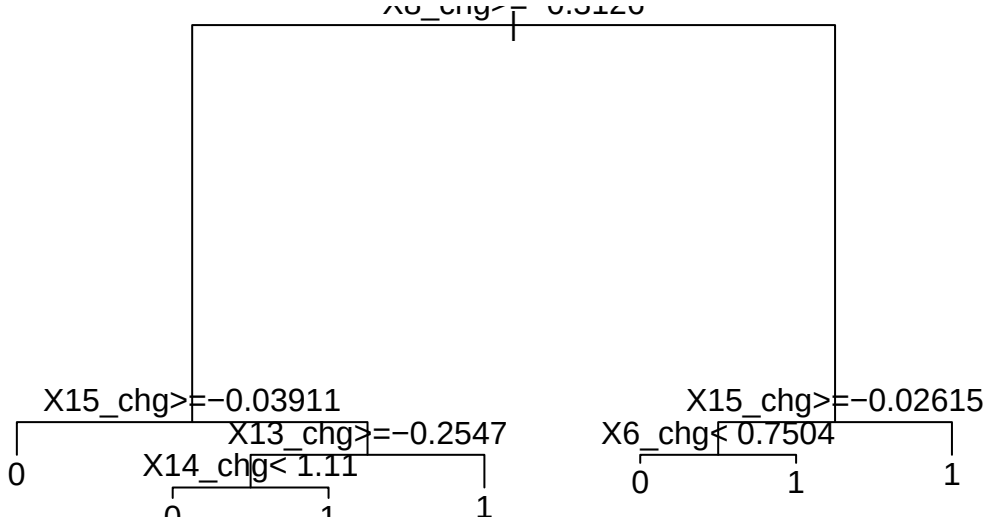
Final structure: bankrupt dummy: 0=no, 1=yes (at the last year before bankrupcy) X1_chg to X18_chg: yearly change in item of financial statments (used 3-years moving average)

```
# set train and test sets (70% train / 30% test)
set.seed(1923)
train <- sample(1:nrow(df), nrow(df) * 0.7)
train_data <- df[train, ]
test_data <- df[-train, ]
train_data$bankrupt <- as.factor(train_data$bankrupt)

# Due to the very few positive tag (=bankrupts) I've used an overweigt for the bankrupt=1 cases
weights <- ifelse(train_data$bankrupt == "0",
                        (1/table(train_data$bankrupt)[1]) * 0.5,
                        (1/table(train_data$bankrupt)[2]) * 0.5)

# Here tried to use oversampling with ROSE library
# First made a balanced sample with 50% bankrupt flag then run the models on
# this dataset. The results was not better hence I went back to the original
# approach.
#over <- ovun.sample(bankrupt~., data = train_data, method = "over", N = sum(train_data$bankrupt == 0)*
#table(over$bankrupt)

tree_model <- rpart(bankrupt ~ ., data = train_data, weights = weights, method = "class")
print(tree_model)
```

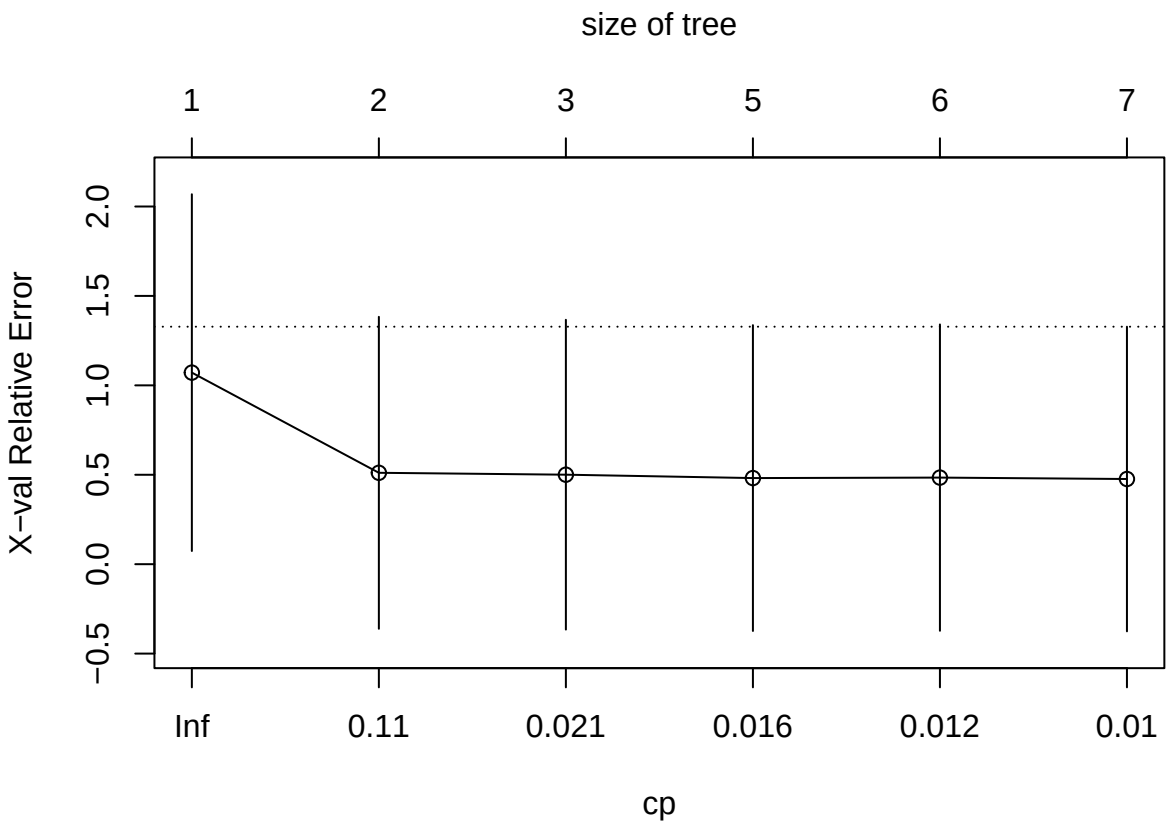
```
## n= 27750
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 27750 0.500000000 0 (0.5000000 0.5000000)
##    2) X8_chg>=-0.312642 21245 0.128289500 0 (0.7503616 0.2496384)
##      4) X15_chg>=-0.0391139 14517 0.039473680 0 (0.8699390 0.1300610) *
##      5) X15_chg< -0.0391139 6728 0.088815790 0 (0.5778717 0.4221283)
##       10) X13_chg>=-0.2547017 5549 0.049342110 0 (0.6708002 0.3291998)
##         20) X14_chg< 1.110454 5009 0.032894740 0 (0.7342533 0.2657467) *
##         21) X14_chg>=1.110454 540 0.009655323 1 (0.3698976 0.6301024) *
##       11) X13_chg< -0.2547017 1179 0.021041320 1 (0.3477042 0.6522958) *
##    3) X8_chg< -0.312642 6505 0.114388300 1 (0.2353189 0.7646811)
##      6) X15_chg>=-0.0261511 1646 0.018092110 0 (0.6221196 0.3778804)
```

```
##       12) X6_chg< 0.7504186 1295 0.006578947 0 (0.7814148 0.2185852) *
##       13) X6_chg>=0.7504186 351 0.006266851 1 (0.3524661 0.6475339) *
##      7) X15_chg< -0.0261511 4859 0.084602490 1 (0.1930590 0.8069410) *
```

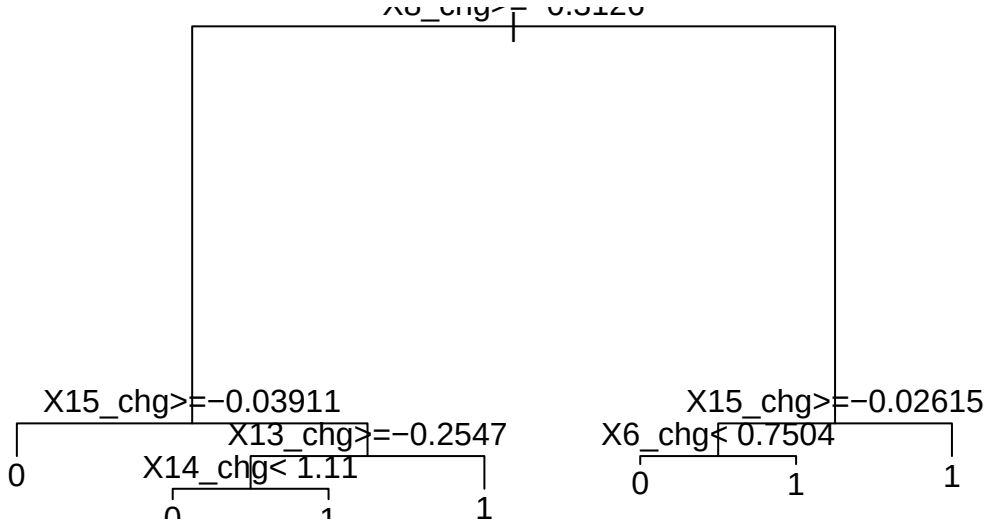
```
# show tree structure
plot(tree_model)
text(tree_model)
```




```
# use the rpart.control function to see whether pruning the tree will improve performance.
cv_model <- rpart(bankrupt ~ ., data = train_data, weights = weights, method = "class",
                  control = rpart.control(cp = 0.01, minsplit = 10, xval = 10))
plotcp(cv_model)
```




```
# prune the tree
pruned_model <- prune(tree_model, cp = tree_model$cptable[which.min(tree_model$cptable[, "xerror"]), "CP
plot(pruned_model)
text(pruned_model)
```




```
# prediction
predictions <- predict(pruned_model, newdata = test_data, type = "class")
actual_values <- as.factor(test_data$bankrupt)
confusion_matrix_tree <- confusionMatrix(predictions, actual_values)
print(confusion_matrix_tree)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 8864   34
##          1 2909   87
##
##                Accuracy : 0.7526
##                  95% CI : (0.7447, 0.7603)
##     No Information Rate : 0.9898
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.037
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.75291
##             Specificity : 0.71901
##          Pos Pred Value : 0.99618
##          Neg Pred Value : 0.02904
```

```
##              Prevalence : 0.98983
##          Detection Rate : 0.74525
##    Detection Prevalence : 0.74811
##       Balanced Accuracy : 0.73596
##
##        'Positive' Class : 0
##
```

```
bag_model <- randomForest(bankrupt ~ ., data = train_data,
                          mtry = 18, weights = weights,
                          importance = TRUE)
print(bag_model)
```

```
##
## Call:
##  randomForest(formula = bankrupt ~ ., data = train_data, mtry = 18,      weights = weights, importan
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 18
##
##         OOB estimate of  error rate: 0.35%
## Confusion matrix:
##       0  1 class.error
## 0 27349 97 0.003534213
## 1     0  0         NaN
```

```
# mtry = 18 indicates that all 18 predictors should be considered for each split of the tree

predictions <- as.factor(predict(bag_model , newdata = test_data))
confusion_matrix_bagging <- confusionMatrix(predictions, actual_values)
print(confusion_matrix_bagging)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 11730   110
##          1    43    11
##
##                Accuracy : 0.9871
##                  95% CI : (0.9849, 0.9891)
##     No Information Rate : 0.9898
##     P-Value [Acc > NIR] : 0.9979
##
##                   Kappa : 0.1202
##
##  Mcnemar's Test P-Value : 9.513e-08
##
##             Sensitivity : 0.99635
##             Specificity : 0.09091
##          Pos Pred Value : 0.99071
##          Neg Pred Value : 0.20370
##              Prevalence : 0.98983
```

```
##          Detection Rate : 0.98621
##    Detection Prevalence : 0.99546
##       Balanced Accuracy : 0.54363
##
##        'Positive' Class : 0
##
```

```
# By default, randomForest() uses p/3 variables when building a random forest of regression trees
# The random forest function inputs are the same as bagging, the difference is the called output
# We use mtry = 6.
rf_model <- randomForest(bankrupt ~ ., data = train_data,
                         weights = weights, importance = TRUE)
print(rf_model)
```

```
##
## Call:
##  randomForest(formula = bankrupt ~ ., data = train_data, weights = weights,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 0.04%
## Confusion matrix:
##       0  1  class.error
## 0 27434 12 0.0004372222
## 1     0  0          NaN
```

```
predictions <- as.factor(predict(rf_model , newdata = test_data))
confusion_matrix_rf <- confusionMatrix(predictions, actual_values)
print(confusion_matrix_rf)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 11760   112
##          1    13     9
##
##                Accuracy : 0.9895
##                  95% CI : (0.9875, 0.9912)
##     No Information Rate : 0.9898
##     P-Value [Acc > NIR] : 0.6641
##
##                   Kappa : 0.1231
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.99890
##             Specificity : 0.07438
##          Pos Pred Value : 0.99057
##          Neg Pred Value : 0.40909
##              Prevalence : 0.98983
##          Detection Rate : 0.98873
```

```
##    Detection Prevalence : 0.99815
##       Balanced Accuracy : 0.53664
##
##        'Positive' Class : 0
##
```

```
boost_model <- gbm(as.numeric("1"==bankrupt) ~ ., data = train_data, weights = weights,
                   distribution = "bernoulli",
                   n.trees = 5000, interaction.depth = 4)
print(boost_model)
```

```
## gbm(formula = as.numeric("1" == bankrupt) ~ ., distribution = "bernoulli",
##     data = train_data, weights = weights, n.trees = 5000, interaction.depth = 4)
## A gradient boosted model with bernoulli loss function.
## 5000 iterations were performed.
## There were 18 predictors of which 17 had non-zero influence.
```

```
predictions <- as.factor(predict(boost_model , newdata =test_data, n.trees = 5000, type = "response"))
binary_predictions <- ifelse(as.numeric(as.character(predictions)) > 0.5, 1, 0)
sum(binary_predictions)
```

```
## [1] 64
```

```
confusion_matrix_boost<- confusionMatrix(as.factor(binary_predictions), actual_values)
print(confusion_matrix_boost)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 11721   109
##          1    52    12
##
##                Accuracy : 0.9865
##                  95% CI : (0.9842, 0.9885)
##     No Information Rate : 0.9898
##     P-Value [Acc > NIR] : 0.9998
##
##                   Kappa : 0.1236
##
##  Mcnemar's Test P-Value : 1.018e-05
##
##             Sensitivity : 0.99558
##             Specificity : 0.09917
##          Pos Pred Value : 0.99079
##          Neg Pred Value : 0.18750
##              Prevalence : 0.98983
##          Detection Rate : 0.98545
##    Detection Prevalence : 0.99462
##       Balanced Accuracy : 0.54738
##
##        'Positive' Class : 0
##
```

```r
# threshold interations
sensitivity_values <- c()
specificity_values <- c()

for (iter in seq(0, 1, by = 0.05)) {
  predictions <- as.factor(ifelse(predict(boost_model, newdata = test_data,
                                          n.trees = 5000, type = "response") > iter, 1, 0))
  confusion_matrix_boost <- confusionMatrix(predictions, actual_values)
  sensitivity_values <- c(sensitivity_values, confusion_matrix_boost$byClass["Sensitivity"])
  specificity_values <- c(specificity_values, confusion_matrix_boost$byClass["Specificity"])
}
```

```
## Warning in confusionMatrix.default(predictions, actual_values): Levels are not
## in the same order for reference and data. Refactoring data to match.

## Warning in confusionMatrix.default(predictions, actual_values): Levels are not
## in the same order for reference and data. Refactoring data to match.
```

```r
plot(seq(0, 1, by = 0.05), sensitivity_values, type = "l", col = "blue", ylim = c(0, 1),
     xlab = "Threshold", ylab = "Érték", main = "Specificity and sensitivity as function of threshold")
lines(seq(0, 1, by = 0.05), specificity_values, type = "l", col = "red")
legend("topright", legend = c("Sensitivity", "Specificity"), col = c("blue", "red"), lty = 1)
```