Lab 4 memo

Team: Nick Carroll, Xinyu Li, Xiaoqian Wang, Dongdong Li

Exercise1:

Part 1:

Question1:

‘most overfit’ rank: fit\_additive, fit\_over, fit\_selected, fit\_caps

‘most underfit’ rank: fit\_caps, fit\_selected, fit\_over, fit\_additive

Question2:

100 folds

> fit\_caps\_100

[1] 0.2173200 0.2171504 0.2166034 0.2166121 0.2165851 0.2173234 0.2165259 0.2175809

[9] 0.2181506 0.2167158 0.2171325 0.2165057 0.2171381 0.2165902 0.2170547 0.2175147

[17] 0.2163489 0.2173014 0.2165205 0.2169920 0.2176114 0.2168258 0.2169635 0.2169214

[25] 0.2180942 0.2175062 0.2167271 0.2167262 0.2173089 0.2170045 0.2163191 0.2170435

[33] 0.2162522 0.2165654 0.2164695 0.2169139 0.2166192 0.2168622 0.2163525 0.2167005

[41] 0.2165991 0.2170467 0.2170075 0.2166019 0.2163182 0.2167533 0.2168971 0.2174494

[49] 0.2171786 0.2171607 0.2172076 0.2172397 0.2165514 0.2162041 0.2166088 0.2169924

[57] 0.2181953 0.2162455 0.2171395 0.2180683 0.2171701 0.2170705 0.2163819 0.2166565

[65] 0.2165166 0.2167406 0.2167898 0.2178925 0.2172771 0.2166971 0.2170988 0.2164482

[73] 0.2162108 0.2164096 0.2176077 0.2165978 0.2167658 0.2173908 0.2176050 0.2164597

[81] 0.2165950 0.2170509 0.2172564 0.2171874 0.2162400 0.2169310 0.2172625 0.2165792

[89] 0.2166397 0.2176039 0.2170738 0.2166817 0.2165427 0.2163949 0.2170156 0.2168596

[97] 0.2164245 0.2166324 0.2170335 0.2174051

> fit\_selected\_100

[1] 0.1582888 0.1571544 0.1585605 0.1581942 0.1580149 0.1592000 0.1589748 0.1577776

[9] 0.1588159 0.1588160 0.1582445 0.1587439 0.1608626 0.1580618 0.1614854 0.1607870

[17] 0.1595542 0.1599211 0.1587221 0.1582634 0.1588308 0.1600115 0.1592589 0.1589581

[25] 0.1581354 0.1577451 0.1585539 0.1587493 0.1579783 0.1580054 0.1602008 0.1572341

[33] 0.1584385 0.1587760 0.1578570 0.1575128 0.1616896 0.1587921 0.1576419 0.1600603

[41] 0.1570881 0.1572271 0.1568971 0.1578023 0.1581108 0.1580827 0.1586558 0.1592670

[49] 0.1587735 0.1570748 0.1588612 0.1609985 0.1592473 0.1591838 0.1605863 0.1581940

[57] 0.1580450 0.1580322 0.1584898 0.1569496 0.1589853 0.1593965 0.1591430 0.1586963

[65] 0.1579272 0.1582487 0.1570451 0.1581942 0.1598155 0.1591807 0.1578910 0.1589119

[73] 0.1586943 0.1572289 0.1576896 0.1572333 0.1618962 0.1588855 0.1592349 0.1580922

[81] 0.1608039 0.1597184 0.1604026 0.1585353 0.1599348 0.1586093 0.1598558 0.1590555

[89] 0.1577749 0.1571154 0.1598765 0.1583290 0.1585722 0.1581037 0.1593300 0.1576796

[97] 0.1588774 0.1591829 0.1589361 0.1589110

> fit\_additive\_100

[1] 0.08438307 0.08381140 0.08072048 0.08178671 0.07666665 0.08723421 0.08519406

[8] 0.08278619 0.08181809 0.08823148 0.08328388 0.08589054 0.08740136 0.08394367

[15] 0.07946037 0.08746940 0.09365969 0.08442650 0.07813296 0.08632906 0.08756927

[22] 0.08694646 0.08432024 0.08668660 0.08245161 0.08404971 0.08565575 0.08673117

[29] 0.07678036 0.08256486 0.08025157 0.08242984 0.08372620 0.08911763 0.08976960

[36] 0.08552723 0.08601633 0.08363780 0.08654311 0.08166768 0.08534227 0.07961880

[43] 0.08864602 0.07487595 0.08076412 0.08070717 0.08645003 0.08224547 0.08950460

[50] 0.08353729 0.07990592 0.08345413 0.08496074 0.08239121 0.08341319 0.08152330

[57] 0.08107926 0.08317148 0.08558267 0.08207663 0.08351621 0.08275116 0.08438724

[64] 0.08474317 0.08025488 0.08514701 0.08167730 0.08393424 0.08019858 0.07565401

[71] 0.09057229 0.09186795 0.08052689 0.08671281 0.08732976 0.08358636 0.08497946

[78] 0.08086167 0.09113001 0.08324848 0.08701816 0.08249047 0.08487710 0.08037997

[85] 0.08495234 0.08208786 0.07813877 0.09302671 0.08568212 0.08223101 0.08449952

[92] 0.10369282 0.08267234 0.08012691 0.08143224 0.08391805 0.07995951 0.07797367

[99] 0.08351122 0.08509253

> fit\_over\_100

[1] 0.1280000 0.1334356 0.1350000 0.1570000 0.1340000 0.1360000 0.1380000 0.1510000

[9] 0.1330000 0.1540000 0.1380000 0.1360000 0.1550000 0.1530000 0.1250000 0.1507340

[17] 0.1510000 0.1570000 0.1530000 0.1460000 0.1450000 0.1390000 0.1380000 0.1430000

[25] 0.1290000 0.1450000 0.1520000 0.1240000 0.1300000 0.1310000 0.1490000 0.1320000

[33] 0.1230000 0.1470000 0.1680000 0.1410000 0.1380000 0.1440000 0.1540000 0.1197591

[41] 0.1510000 0.1265626 0.1580000 0.1510000 0.1410000 0.1340000 0.1360000 0.1380000

[49] 0.1480000 0.1540000 0.1310000 0.1520000 0.1430000 0.1340000 0.1540000 0.1540000

[57] 0.1420000 0.1520000 0.1520000 0.1590000 0.1310000 0.1417497 0.1360000 0.1410000

[65] 0.1300000 0.1340000 0.1450000 0.1400000 0.1260000 0.1410000 0.1390000 0.1400000

[73] 0.1340000 0.1370000 0.1630000 0.1390000 0.1360000 0.1450000 0.1430000 0.1400000

[81] 0.1510000 0.1540000 0.1590000 0.1380000 0.1340000 0.1392271 0.1610000 0.1430000

[89] 0.1640000 0.1310000 0.1480000 0.1350000 0.1450000 0.1420000 0.1390000 0.1480000

[97] 0.1250000 0.1380000 0.1450000 0.1580000

> mean(fit\_caps\_100)

[1] 0.2169212

> mean(fit\_selected\_100)

[1] 0.1587243

> mean(fit\_additive\_100)

[1] 0.08401166

> mean(fit\_over\_100)

[1] 0.1424547

Base on the data above the ranks are still:

‘most overfit’ rank: fit\_additive, fit\_over, fit\_selected, fit\_caps

‘most underfit’ rank: fit\_caps, fit\_selected, fit\_over, fit\_additive

Part 2:

Question1:

Fit\_additive

> (conf\_mat\_50 = make\_conf\_mat(predicted = spam\_tst\_pred, actual = spam\_tst$type))

Actual

predicted nonspam spam

nonspam 2057 157

spam 127 1260

> table(spam\_tst$type) / nrow(spam\_tst)

nonspam spam

0.6064982 0.3935018

Fit\_selected

> (conf\_mat\_50 = make\_conf\_mat(predicted = spam\_tst\_pred, actual = spam\_tst$type))

actual

predicted nonspam spam

nonspam 2073 615

spam 111 802

> table(spam\_tst$type) / nrow(spam\_tst)

nonspam spam

0.6064982 0.3935018

Fit\_caps

> (conf\_mat\_50 = make\_conf\_mat(predicted = spam\_tst\_pred, actual = spam\_tst$type))

actual

predicted nonspam spam

nonspam 2022 1066

spam 162 351

> table(spam\_tst$type) / nrow(spam\_tst)

nonspam spam

0.6064982 0.3935018

Fit\_over

> (conf\_mat\_50 = make\_conf\_mat(predicted = spam\_tst\_pred, actual = spam\_tst$type))

actual

predicted nonspam spam

nonspam 1725 103

spam 459 1314

> table(spam\_tst$type) / nrow(spam\_tst)

nonspam spam

0.6064982 0.3935018

Question4: We believe that the fit\_selected is the best model cause it not only not too overfit to make a useful prediction but also follows the model.

Exercise2:

3. Discuss the interpretation of the coefficients in your model

A screenshot of text

Description automatically generated

* Age had a positive coefficient (2.03\*10-2), which means an additional year of age expect the chance y=yes increase by 2.03\*10-2. Also, it is a statistically significant since probability(>z) is 0.000883 < 0.005.
* Education secondary had a positive coefficient (2.539\*10-1), which means an additional education secondary level expect the chance of y= yes increase by 2.539\*10-1. Also, it is not a statistically significant since probability(>z) is 0.21702 > 0.005.
* Education tertiary had a positive coefficient (4.829\*10-1), which means an additional education tertiary level expect the chance of y=yes increase by 4.829\*10-1. Also, t is not a statistically significant since the probability(>z) is 0.025183 > 0.005.
* Balance has a positive coefficient (6.235\*10-6), which means an additional balance expected the chance of y=yes increase by 6.235\*10-6. Also, it is not a statistically significant since the probability(>z) is 0.759543 > 0.005.
* Default had a negative coefficient (-3.211\*10-1), which means an additional default expected the chance of y=no increase by -3.211\*10-1. also, it is not a statistically significant since the probability(>z) is 0.545268 > 0.005.