Data 410 Project Rough Draft

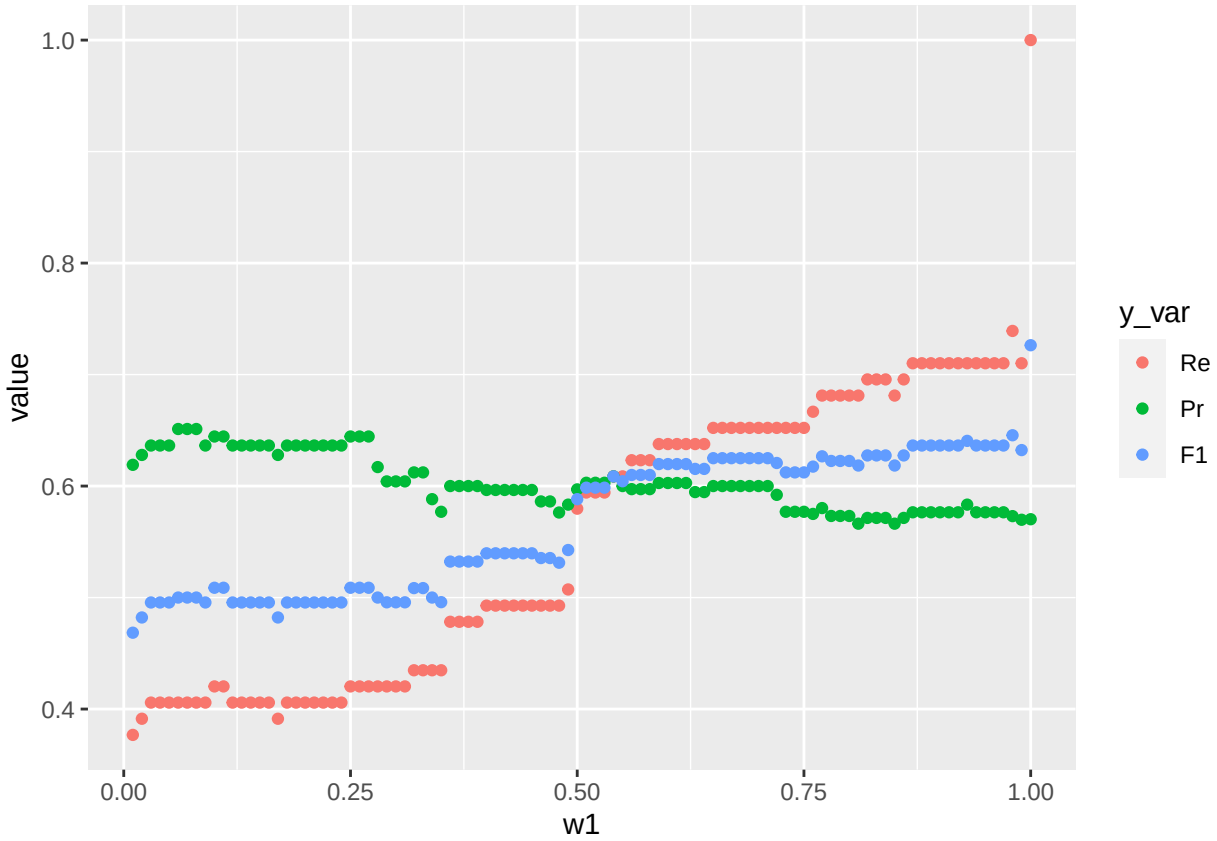
Daniel Krasnov, Keiran Malott, Ross Cooper

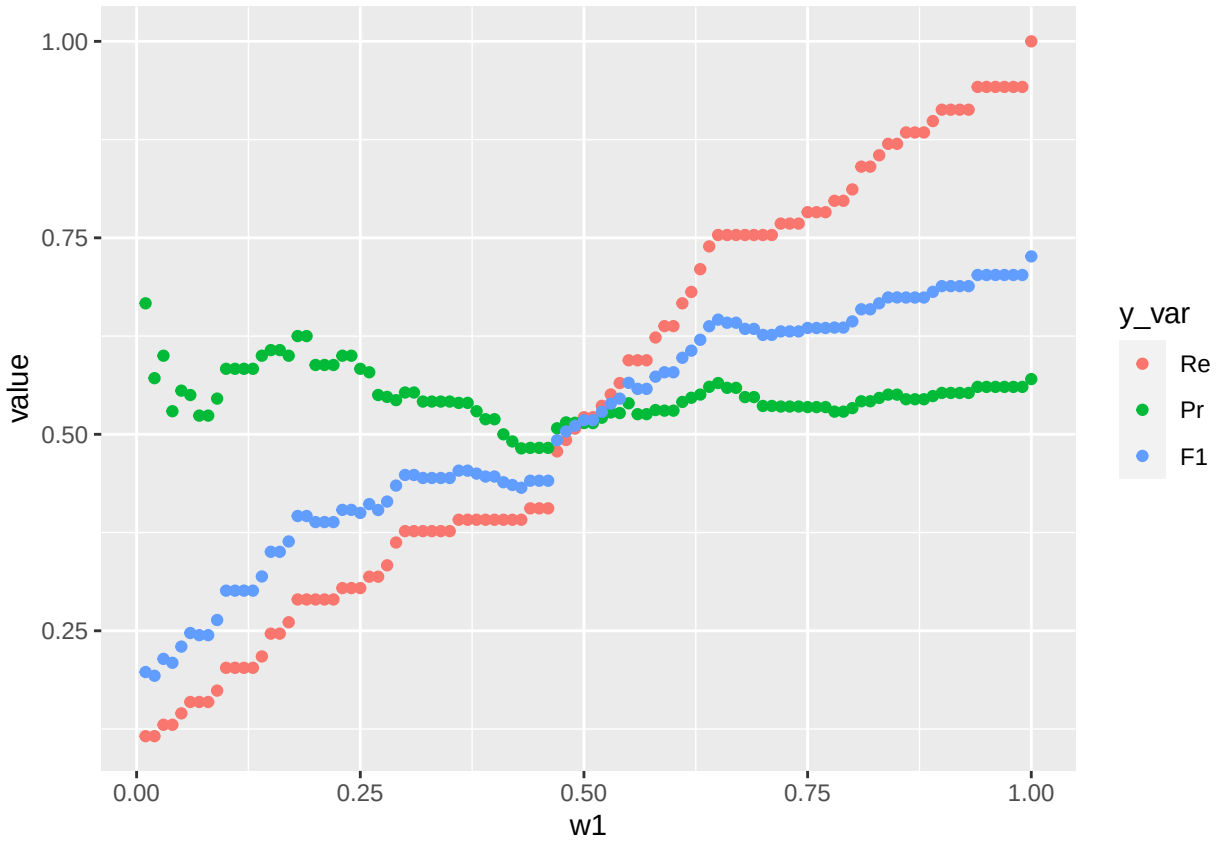
2023-04-07

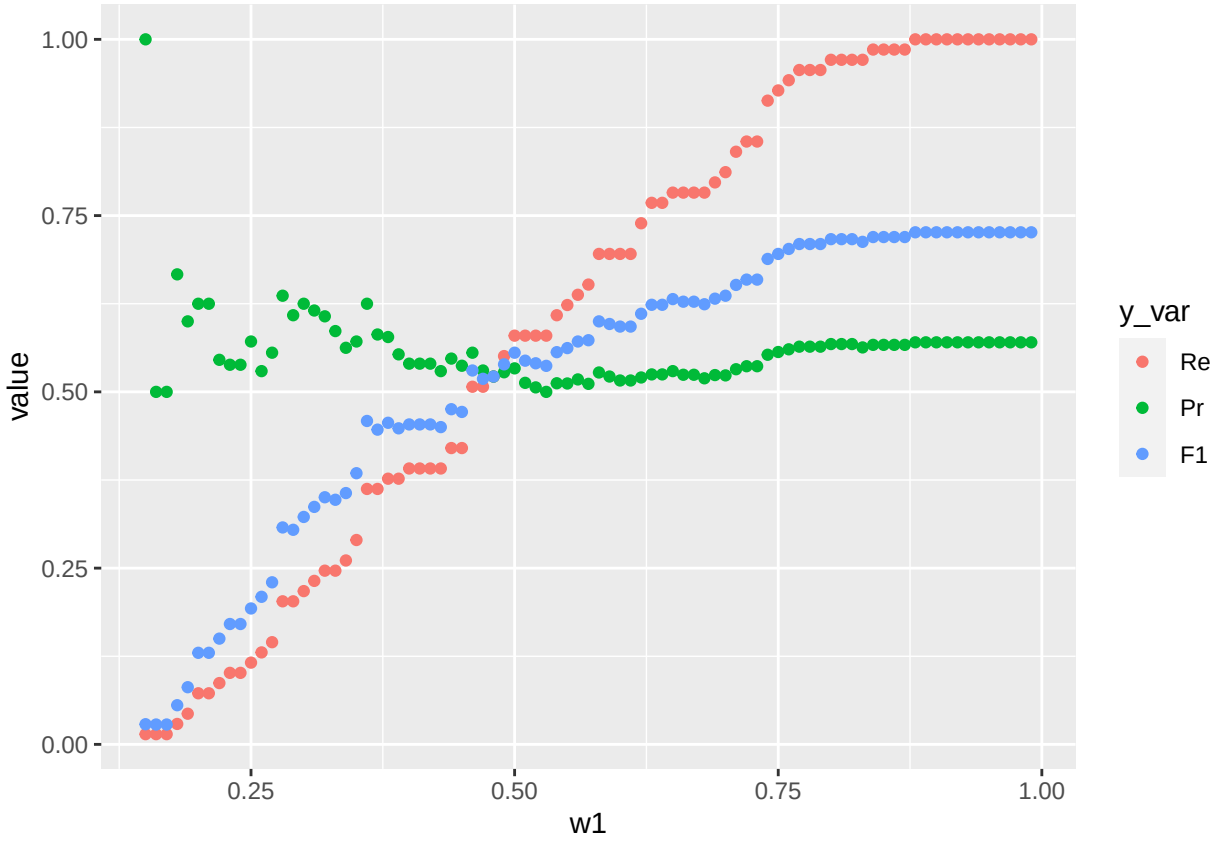
Contents

```
## INFO  [19:27:57.147] epoch 1, loss 0.2237
## INFO  [19:27:57.183] epoch 2, loss 0.0890
## INFO  [19:27:57.193] epoch 3, loss 0.0430
## INFO  [19:27:57.198] epoch 4, loss 0.0212
## INFO  [19:27:57.203] epoch 5, loss 0.0141
## INFO  [19:27:57.209] epoch 6, loss 0.0099
## INFO  [19:27:57.213] epoch 7, loss 0.0071
## INFO  [19:27:57.218] epoch 8, loss 0.0052
## INFO  [19:27:57.227] epoch 9, loss 0.0039
## INFO  [19:27:57.234] epoch 10, loss 0.0030
```

```
## INFO  [18:01:13.625] epoch 1, loss 0.2218
## INFO  [18:01:13.637] epoch 2, loss 0.0837
## INFO  [18:01:13.650] epoch 3, loss 0.0439
## INFO  [18:01:13.663] epoch 4, loss 0.0213
## INFO  [18:01:13.673] epoch 5, loss 0.0134
## INFO  [18:01:13.684] epoch 6, loss 0.0093
## INFO  [18:01:13.694] epoch 7, loss 0.0067
## INFO  [18:01:13.705] epoch 8, loss 0.0049
## INFO  [18:01:13.716] epoch 9, loss 0.0037
## INFO  [18:01:13.726] epoch 10, loss 0.0028
```




1.00
0.75
0.50
0.25
value
0.00
0.25
0.50
0.75
1.00
w1
y_var
Re
Pr
F1




```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = training_fit)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2342  -0.4333   0.0034   0.4636   3.6833
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2559.2392  1934.1075  -1.323 0.185764
## X1            -69.8265    29.1250  -2.397 0.016509 *
## X2             22.9997    30.5300   0.753 0.451242
## X3             22.9678    27.5994   0.832 0.405305
## X4            -60.3060    32.0211  -1.883 0.059657 .
## X5              7.9076    27.4179   0.288 0.773033
## X6           -118.8379    31.4383  -3.780 0.000157 ***
## X7             84.2241    29.7426   2.832 0.004629 **
## X8           -102.0366    30.6789  -3.326 0.000881 ***
## X9            -54.1699    30.8491  -1.756 0.079095 .
## X10           -47.8978    26.2988  -1.821 0.068562 .
## X11            22.7691    27.3930   0.831 0.405859
## X12           -99.8844    34.0597  -2.933 0.003361 **
## X13            12.3567    30.6968   0.403 0.687286
## X14            22.2789    30.1181   0.740 0.459472
## X15           -70.2127    26.5710  -2.642 0.008231 **
```

```
## X16           19.1698    30.0990   0.637 0.524196
## X17          -10.6338    24.5234  -0.434 0.664565
## X18           12.0193    30.2311   0.398 0.690940
## X19          -22.7283    32.2056  -0.706 0.480359
## X20           13.6993    29.8642   0.459 0.646435
## X21           30.8821    29.8674   1.034 0.301148
## X22          -10.5342    34.2438  -0.308 0.758370
## X23          -17.8965    27.6876  -0.646 0.518039
## X24           -8.1658    32.1033  -0.254 0.799218
## X25          -74.1264    34.6810  -2.137 0.032567 *
## X26          -99.3026    52.9785  -1.874 0.060876 .
## X27          -32.7864    29.5302  -1.110 0.266884
## X28          -38.8960    35.7619  -1.088 0.276755
## X29          -12.5160    32.4301  -0.386 0.699544
## X30           11.4689    29.4718   0.389 0.697168
## X31          -61.6006    29.3451  -2.099 0.035801 *
## X32          -78.8929    33.8445  -2.331 0.019751 *
## X33          -10.3685    28.9344  -0.358 0.720085
## X34          -21.7151    35.0711  -0.619 0.535802
## X35           42.7749    27.9236   1.532 0.125559
## X36          -24.3078    25.8256  -0.941 0.346588
## X37          -34.5420    29.0019  -1.191 0.233643
## X38          -28.0230    30.1176  -0.930 0.352137
## X39           90.1622    43.8399   2.057 0.039722 *
## X40         -122.0489    33.3550  -3.659 0.000253 ***
## X41          -32.7833    26.9423  -1.217 0.223683
## X42            7.9077    32.2364   0.245 0.806221
## X43          -68.6886    31.8762  -2.155 0.031173 *
## X44          -31.9918    30.0773  -1.064 0.287487
## X45          100.5566    35.2040   2.856 0.004285 **
## X46          -41.1031    32.8207  -1.252 0.210441
## X47            4.5598    25.0551   0.182 0.855589
## X48           71.1981    31.4444   2.264 0.023559 *
## X49          -49.7616    27.6393  -1.800 0.071799 .
## X50           29.2174    30.7327   0.951 0.341759
## X51           14.7138    30.6162   0.481 0.630807
## X52           18.4013    37.5813   0.490 0.624390
## X53           -8.7796    35.3319  -0.248 0.803755
## X54           88.4910    32.1621   2.751 0.005934 **
## X55          -34.3052    30.9367  -1.109 0.267480
## X56           -1.5789    36.2869  -0.044 0.965295
## X57          -35.3280    32.0595  -1.102 0.270483
## X58           36.5888    30.8036   1.188 0.234908
## X59           17.2426    27.5446   0.626 0.531324
## X60          -41.7165    30.9977  -1.346 0.178369
## X61          -99.5842    35.8648  -2.777 0.005492 **
## X62            1.9662    35.9496   0.055 0.956382
## X63            8.2065    30.2856   0.271 0.786413
## X64          -24.8369    30.0993  -0.825 0.409278
## X65           16.6265    27.8858   0.596 0.551019
## X66          -17.7462    32.9302  -0.539 0.589953
## X67           85.7834    39.9678   2.146 0.031848 *
## X68          -36.1401    29.5751  -1.222 0.221717
## X69           26.5559    25.1644   1.055 0.291290
```

```
## X70           15.7676    31.5299   0.500 0.617016
## X71           -9.2241    30.0247  -0.307 0.758678
## X72          -37.7073    32.6657  -1.154 0.248361
## X73          -23.3708    32.8448  -0.712 0.476742
## X74          131.0104    35.2140   3.720 0.000199 ***
## X75           75.8644    39.2142   1.935 0.053038 .
## X76           31.2431    29.3769   1.064 0.287545
## X77          -55.1523    35.9410  -1.535 0.124901
## X78           43.4288    26.8683   1.616 0.106017
## X79           14.9516    26.8123   0.558 0.577090
## X80           12.3569    34.3426   0.360 0.718987
## X81          -24.8884    35.5687  -0.700 0.484098
## X82           50.8537    27.2418   1.867 0.061936 .
## X83          -11.2827    32.3112  -0.349 0.726948
## X84          -23.5222    28.1816  -0.835 0.403907
## X85          -47.8629    30.7377  -1.557 0.119438
## X86           37.3308    31.6645   1.179 0.238418
## X87           33.1714    30.7414   1.079 0.280568
## X88           14.8955    26.9634   0.552 0.580652
## X89           27.5104    33.2650   0.827 0.408232
## X90           -9.1011    23.8948  -0.381 0.703289
## X91          -41.6122    25.5634  -1.628 0.103566
## X92           85.9110    28.4032   3.025 0.002489 **
## X93          -32.7740    37.7702  -0.868 0.385548
## X94            6.5044    28.3207   0.230 0.818347
## X95          -84.7743    33.3820  -2.540 0.011100 *
## X96          -15.1698    28.0512  -0.541 0.588652
## X97            7.1077    28.8937   0.246 0.805687
## X98          -39.8421    29.7256  -1.340 0.180138
## X99           59.5924    30.9462   1.926 0.054145 .
## X100          93.6876    31.7734   2.949 0.003192 **
## X101          42.3018    33.8317   1.250 0.211168
## X102         -67.7467    29.1526  -2.324 0.020133 *
## X103         -38.1984    27.5832  -1.385 0.166102
## X104         -43.4435    29.7495  -1.460 0.144205
## X105        -101.0569    41.7621  -2.420 0.015528 *
## X106         100.2445    28.9996   3.457 0.000547 ***
## X107         -51.6920    27.4170  -1.885 0.059376 .
## X108          -5.4574    30.2987  -0.180 0.857059
## X109         -21.1796    33.8423  -0.626 0.531424
## X110         -25.6506    28.8080  -0.890 0.373251
## X111         -39.2775    29.6935  -1.323 0.185915
## X112         -46.8147    37.0347  -1.264 0.206202
## X113         -18.5296    31.7404  -0.584 0.559364
## X114         -28.3563    28.5598  -0.993 0.320771
## X115         -95.7032    30.0163  -3.188 0.001431 **
## X116          14.8734    27.1276   0.548 0.583504
## X117          -4.7937    31.4829  -0.152 0.878979
## X118         -94.6361    36.6835  -2.580 0.009886 **
## X119           2.9604    26.6935   0.111 0.911695
## X120         -30.3433    36.3338  -0.835 0.403646
## X121          73.6990    32.3109   2.281 0.022553 *
## X122         -21.7009    33.1770  -0.654 0.513051
## X123         -44.0019    28.8601  -1.525 0.127344
```

```
## X124            4.2356    39.7727   0.106 0.915190
## X125          -16.7395    30.4416  -0.550 0.582395
## X126          -10.8914    40.7406  -0.267 0.789212
## X127          -98.7873    34.6679  -2.850 0.004378 **
## X128           17.5726    30.4130   0.578 0.563399
## X129          -51.5909    27.8755  -1.851 0.064203 .
## X130          -12.2402    33.9885  -0.360 0.718753
## X131          -41.9436    29.9833  -1.399 0.161843
## X132          -65.4529    40.8593  -1.602 0.109176
## X133           28.9322    30.9442   0.935 0.349798
## X134          -98.0249    65.5141  -1.496 0.134591
## X135            0.1853    29.4185   0.006 0.994973
## X136           26.0617    29.4216   0.886 0.375725
## X137           81.6910    34.1885   2.389 0.016875 *
## X138           45.7415    39.4424   1.160 0.246170
## X139         -104.0432    31.9716  -3.254 0.001137 **
## X140           26.5768    38.6470   0.688 0.491654
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 648.37  on 467  degrees of freedom
## Residual deviance: 311.05  on 327  degrees of freedom
## AIC: 593.05
##
## Number of Fisher Scoring iterations: 8
```

```
## Importance of components:
##                             PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     0.09519 0.05838 0.05538 0.04012 0.03662 0.03591 0.03472
## Proportion of Variance 0.19463 0.07320 0.06588 0.03458 0.02880 0.02770 0.02589
## Cumulative Proportion  0.19463 0.26782 0.33371 0.36828 0.39709 0.42478 0.45067
##                             PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.03406 0.03224 0.03176 0.03025 0.02864 0.02789 0.02743
## Proportion of Variance 0.02492 0.02232 0.02167 0.01966 0.01762 0.01671 0.01616
## Cumulative Proportion  0.47559 0.49791 0.51958 0.53924 0.55686 0.57357 0.58973
##                            PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     0.02736 0.02629 0.02591 0.02529 0.02466 0.02407 0.02338
## Proportion of Variance 0.01608 0.01485 0.01442 0.01374 0.01306 0.01244 0.01174
## Cumulative Proportion  0.60580 0.62065 0.63507 0.64881 0.66187 0.67431 0.68605
##                            PC22    PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation     0.02291 0.02228 0.02163 0.02151 0.02099 0.02089 0.02077
## Proportion of Variance 0.01127 0.01067 0.01005 0.00994 0.00946 0.00937 0.00926
## Cumulative Proportion  0.69732 0.70799 0.71804 0.72798 0.73744 0.74681 0.75607
##                            PC29    PC30    PC31    PC32    PC33    PC34    PC35
## Standard deviation     0.01998 0.01964 0.01943 0.01894 0.01841 0.01815 0.01758
## Proportion of Variance 0.00857 0.00829 0.00811 0.00770 0.00728 0.00707 0.00664
## Cumulative Proportion  0.76464 0.77293 0.78103 0.78874 0.79602 0.80309 0.80973
##                            PC36    PC37    PC38    PC39   PC40    PC41    PC42
## Standard deviation     0.01730 0.01716 0.01665 0.01635 0.0160 0.01551 0.01520
## Proportion of Variance 0.00643 0.00633 0.00595 0.00574 0.0055 0.00516 0.00496
## Cumulative Proportion  0.81616 0.82249 0.82844 0.83418 0.8397 0.84484 0.84980
##                            PC43    PC44    PC45    PC46    PC47    PC48    PC49
```

```
## Standard deviation     0.01494 0.01491 0.01473 0.01432 0.01419 0.01406 0.01363
## Proportion of Variance 0.00480 0.00477 0.00466 0.00440 0.00433 0.00425 0.00399
## Cumulative Proportion  0.85460 0.85937 0.86403 0.86844 0.87276 0.87701 0.88100
##                           PC50    PC51    PC52    PC53    PC54    PC55    PC56
## Standard deviation     0.01351 0.01337 0.01310 0.01299 0.01275 0.01241 0.01233
## Proportion of Variance 0.00392 0.00384 0.00369 0.00363 0.00349 0.00331 0.00326
## Cumulative Proportion  0.88492 0.88876 0.89244 0.89607 0.89956 0.90287 0.90613
##                           PC57    PC58    PC59    PC60    PC61    PC62    PC63
## Standard deviation     0.01223 0.01214 0.01194 0.01171 0.01166 0.01132 0.01115
## Proportion of Variance 0.00321 0.00317 0.00306 0.00294 0.00292 0.00275 0.00267
## Cumulative Proportion  0.90935 0.91251 0.91558 0.91852 0.92144 0.92419 0.92686
##                           PC64    PC65    PC66    PC67    PC68     PC69
## Standard deviation     0.01112 0.01083 0.01078 0.01045 0.01037 0.009989
## Proportion of Variance 0.00266 0.00252 0.00249 0.00235 0.00231 0.002140
## Cumulative Proportion  0.92952 0.93204 0.93453 0.93688 0.93919 0.941330
##                             PC70     PC71     PC72     PC73     PC74     PC75
## Standard deviation     0.009943 0.009842 0.009704 0.009563 0.009438 0.009407
## Proportion of Variance 0.002120 0.002080 0.002020 0.001960 0.001910 0.001900
## Cumulative Proportion  0.943450 0.945530 0.947560 0.949520 0.951430 0.953330
##                             PC76     PC77    PC78    PC79     PC80     PC81
## Standard deviation     0.009316 0.009029 0.00893 0.00877 0.008642 0.008478
## Proportion of Variance 0.001860 0.001750 0.00171 0.00165 0.001600 0.001540
## Cumulative Proportion  0.955200 0.956950 0.95866 0.96031 0.961920 0.963460
##                             PC82     PC83     PC84     PC85    PC86     PC87
## Standard deviation     0.008352 0.008194 0.008039 0.007925 0.00784 0.007652
## Proportion of Variance 0.001500 0.001440 0.001390 0.001350 0.00132 0.001260
## Cumulative Proportion  0.964960 0.966400 0.967790 0.969140 0.97046 0.971720
##                             PC88     PC89     PC90     PC91     PC92     PC93
## Standard deviation     0.00752 0.007372 0.007245 0.007032 0.006967 0.006871
## Proportion of Variance 0.00121 0.001170 0.001130 0.001060 0.001040 0.001010
## Cumulative Proportion  0.97293 0.974100 0.975220 0.976290 0.977330 0.978340
##                             PC94     PC95     PC96     PC97     PC98     PC99
## Standard deviation     0.006755 0.006685 0.006588 0.006455 0.006394 0.006209
## Proportion of Variance 0.000980 0.000960 0.000930 0.000890 0.000880 0.000830
## Cumulative Proportion  0.979320 0.980280 0.981220 0.982110 0.982990 0.983820
##                            PC100    PC101    PC102   PC103    PC104    PC105
## Standard deviation     0.006143 0.005937 0.005769 0.00569 0.005675 0.005657
## Proportion of Variance 0.000810 0.000760 0.000710 0.00070 0.000690 0.000690
## Cumulative Proportion  0.984630 0.985380 0.986100 0.98679 0.987490 0.988170
##                            PC106    PC107   PC108    PC109    PC110    PC111
## Standard deviation     0.005348 0.005309 0.00522 0.005195 0.005012 0.004949
## Proportion of Variance 0.000610 0.000610 0.00059 0.000580 0.000540 0.000530
## Cumulative Proportion  0.988790 0.989390 0.98998 0.990560 0.991100 0.991620
##                            PC112    PC113    PC114    PC115    PC116   PC117
## Standard deviation     0.004826 0.004788 0.004668 0.004597 0.004547 0.00444
## Proportion of Variance 0.000500 0.000490 0.000470 0.000450 0.000440 0.00042
## Cumulative Proportion  0.992120 0.992620 0.993080 0.993540 0.993980 0.99441
##                           PC118    PC119    PC120   PC121    PC122    PC123
## Standard deviation     0.00434 0.004217 0.004158 0.00407 0.004022 0.003945
## Proportion of Variance 0.00040 0.000380 0.000370 0.00036 0.000350 0.000330
## Cumulative Proportion  0.99481 0.995190 0.995560 0.99592 0.996270 0.996600
##                            PC124   PC125    PC126    PC127    PC128    PC129
## Standard deviation     0.003835 0.00369 0.003615 0.003526 0.003464 0.003339
## Proportion of Variance 0.000320 0.00029 0.000280 0.000270 0.000260 0.000240
```

```
## Cumulative Proportion  0.996920 0.99721 0.997490 0.997760 0.998010 0.998250
##                           PC130    PC131    PC132   PC133    PC134    PC135
## Standard deviation     0.003311 0.003161 0.003127 0.00296 0.002835 0.002788
## Proportion of Variance 0.000240 0.000210 0.000210 0.00019 0.000170 0.000170
## Cumulative Proportion  0.998490 0.998700 0.998910 0.99910 0.999270 0.999440
##                           PC136    PC137   PC138    PC139    PC140
## Standard deviation     0.002673 0.002513 0.00245 0.002245 0.001222
## Proportion of Variance 0.000150 0.000140 0.00013 0.000110 0.000030
## Cumulative Proportion  0.999600 0.999730 0.99986 0.999970 1.000000
##
## Call:
## glm(formula = y ~ ., family = binomial, data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4253  -0.8572   0.2064   0.8641   2.3699
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.09124    0.12206   0.747 0.454767
## PC1           6.29596    1.88878   3.333 0.000858 ***
## PC2          -3.41064    3.22039  -1.059 0.289565
## PC3          12.53847    3.17236   3.952 7.74e-05 ***
## PC4           7.43040    3.66338   2.028 0.042530 *
## PC5          -4.94380    4.97068  -0.995 0.319935
## PC6         -13.46139    4.92851  -2.731 0.006308 **
## PC7          10.15693    4.60385   2.206 0.027371 *
## PC8          10.25764    4.39672   2.333 0.019647 *
## PC9           4.22166    4.29064   0.984 0.325153
## PC10         -1.05162    4.24191  -0.248 0.804202
## PC11          3.60145    4.96480   0.725 0.468208
## PC12         -5.30550    4.54190  -1.168 0.242757
## PC13          2.96653    4.75338   0.624 0.532569
## PC14         -4.70034    5.84443  -0.804 0.421257
## PC15          1.03548    4.36718   0.237 0.812576
## PC16         15.66761    5.02446   3.118 0.001819 **
## PC17          1.75139    4.93852   0.355 0.722861
## PC18         -1.02397    4.90484  -0.209 0.834630
## PC19         -6.50481    5.06890  -1.283 0.199394
## PC20        -21.58493    5.61482  -3.844 0.000121 ***
## PC21          8.56797    5.25480   1.631 0.102995
## PC22          3.09848    5.62192   0.551 0.581536
## PC23          3.20845    5.60907   0.572 0.567315
## PC24         15.80439    6.16125   2.565 0.010314 *
## PC25         -3.34997    5.38726  -0.622 0.534052
## PC26         11.14498    6.36790   1.750 0.080087 .
## PC27          8.07106    5.63032   1.433 0.151715
## PC28          6.24300    6.29045   0.992 0.320974
## PC29        -23.33230    6.39804  -3.647 0.000266 ***
## PC30        -18.95070    6.31109  -3.003 0.002675 **
## PC31          2.95232    6.49842   0.454 0.649603
## PC32         -2.60584    6.37705  -0.409 0.682813
## PC33         -2.06800    6.40770  -0.323 0.746894
```

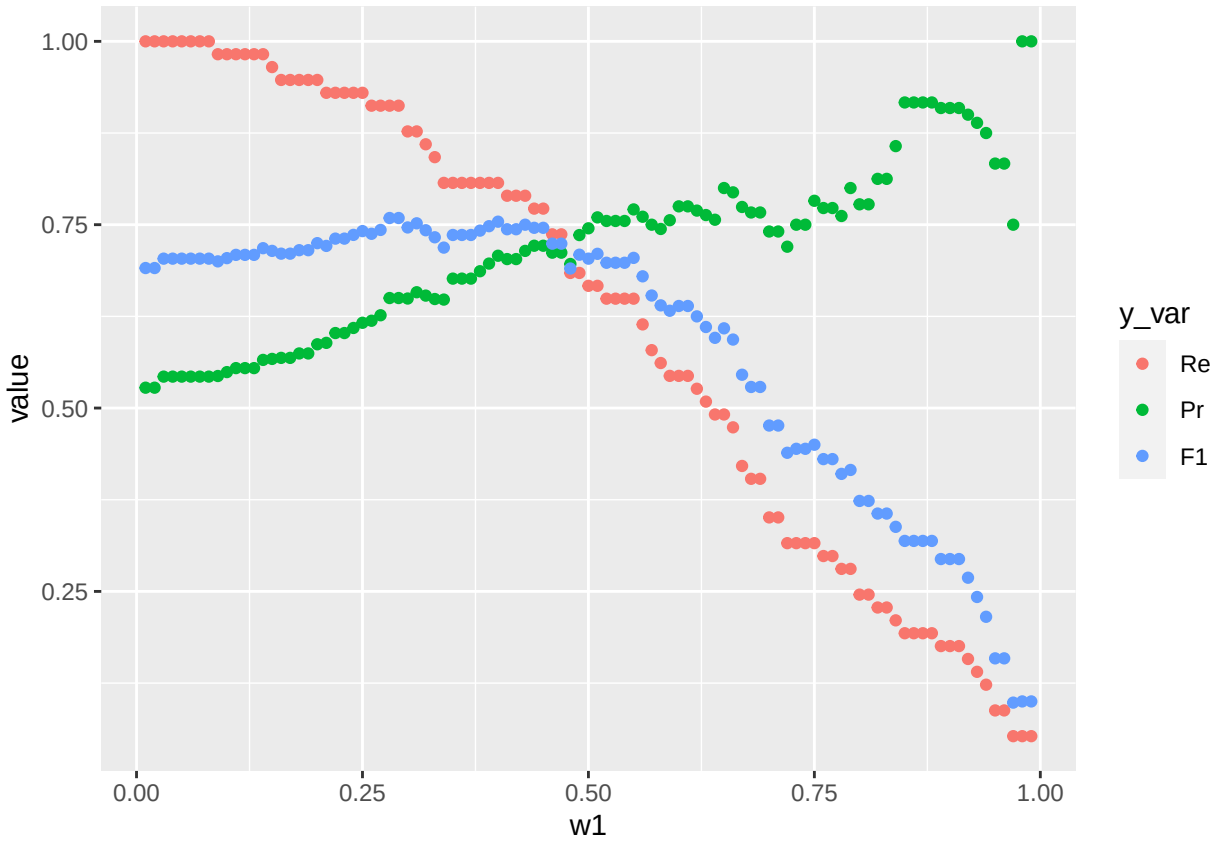
```
## PC34         -1.32021    6.92225  -0.191 0.848745
## PC35         -0.57360    7.25581  -0.079 0.936990
## PC36         -0.03943    6.52724  -0.006 0.995181
## PC37         11.94267    7.32787   1.630 0.103152
## PC38         -1.71410    7.37900  -0.232 0.816309
## PC39         -9.43080    7.51560  -1.255 0.209540
## PC40         13.51025    7.33702   1.841 0.065566 .
## PC41         -5.73622    7.63967  -0.751 0.452745
## PC42         -2.72042    7.85396  -0.346 0.729060
## PC43          9.85039    8.01504   1.229 0.219076
## PC44          0.99267    7.77485   0.128 0.898404
## PC45         31.96314    8.46849   3.774 0.000160 ***
## PC46          7.05721    8.42256   0.838 0.402090
## PC47        -11.76609    8.25520  -1.425 0.154072
## PC48         18.97121    8.31163   2.282 0.022460 *
## PC49         11.26527    9.22468   1.221 0.222006
## PC50        -46.23038    9.58529  -4.823 1.41e-06 ***
## PC51         22.39730    9.00024   2.489 0.012828 *
## PC52         11.19553    8.74784   1.280 0.200614
## PC53        -10.51438    9.00062  -1.168 0.242732
## PC54        -19.80567    9.51523  -2.081 0.037391 *
## PC55         -6.00409    9.50495  -0.632 0.527596
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 648.37  on 467  degrees of freedom
## Residual deviance: 483.32  on 412  degrees of freedom
## AIC: 595.32
##
## Number of Fisher Scoring iterations: 6
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = training_fit,
##     weights = weight_vec)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.71385  -0.32344   0.00212   0.31936   2.62334
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2720.1181  2771.0007  -0.982  0.32628
## X1            -66.0935    41.4844  -1.593  0.11111
## X2             20.7778    44.1777   0.470  0.63812
## X3             29.4210    39.9484   0.736  0.46144
## X4            -65.0726    46.9360  -1.386  0.16562
## X5              5.2488    39.1339   0.134  0.89331
## X6           -122.6022    46.5127  -2.636  0.00839 **
## X7             83.2839    42.5426   1.958  0.05027 .
## X8            -97.7400    44.0700  -2.218  0.02657 *
## X9            -51.3667    43.7870  -1.173  0.24075
```

```
## X10          -47.1508    37.8022  -1.247  0.21229
## X11           23.6838    39.1689   0.605  0.54541
## X12          -88.9470    47.6915  -1.865  0.06217 .
## X13           15.7888    44.2778   0.357  0.72140
## X14           18.0655    43.0839   0.419  0.67499
## X15          -67.0554    37.5472  -1.786  0.07412 .
## X16           19.3415    43.6445   0.443  0.65765
## X17           -7.7845    35.4186  -0.220  0.82604
## X18           15.7984    43.2780   0.365  0.71508
## X19          -26.2101    46.6337  -0.562  0.57409
## X20           19.9021    42.8551   0.464  0.64236
## X21           27.3912    42.8625   0.639  0.52279
## X22           -8.6127    48.7146  -0.177  0.85967
## X23          -19.1283    40.0321  -0.478  0.63278
## X24          -11.8902    45.5551  -0.261  0.79409
## X25          -68.4306    49.1884  -1.391  0.16417
## X26          -99.8224    75.1800  -1.328  0.18425
## X27          -31.1106    41.9998  -0.741  0.45886
## X28          -45.0084    51.3577  -0.876  0.38083
## X29           -9.8868    47.0556  -0.210  0.83358
## X30            7.6856    41.8665   0.184  0.85435
## X31          -57.6065    42.2012  -1.365  0.17224
## X32          -77.9581    48.2339  -1.616  0.10604
## X33          -10.1598    41.5893  -0.244  0.80701
## X34          -18.5599    49.2633  -0.377  0.70636
## X35           45.4653    39.9416   1.138  0.25500
## X36          -26.2122    36.9099  -0.710  0.47760
## X37          -31.1316    41.5961  -0.748  0.45420
## X38          -25.2053    42.4008  -0.594  0.55221
## X39           92.9534    62.6330   1.484  0.13778
## X40         -125.4011    47.9283  -2.616  0.00889 **
## X41          -29.3523    38.7877  -0.757  0.44920
## X42           10.9180    46.4863   0.235  0.81431
## X43          -70.8885    46.1787  -1.535  0.12476
## X44          -29.0173    42.2413  -0.687  0.49212
## X45          102.5089    50.8696   2.015  0.04389 *
## X46          -39.6156    47.2604  -0.838  0.40190
## X47            0.7258    35.7432   0.020  0.98380
## X48           72.1210    44.4900   1.621  0.10500
## X49          -51.5076    39.6102  -1.300  0.19348
## X50           26.5926    44.0599   0.604  0.54614
## X51           17.2133    43.7693   0.393  0.69412
## X52           17.6377    54.3274   0.325  0.74544
## X53          -17.5302    50.4174  -0.348  0.72806
## X54           86.8996    46.0952   1.885  0.05940 .
## X55          -38.2523    44.3116  -0.863  0.38800
## X56            1.6725    52.6511   0.032  0.97466
## X57          -29.9384    46.1285  -0.649  0.51632
## X58           38.9415    44.6893   0.871  0.38355
## X59           17.5451    38.7671   0.453  0.65085
## X60          -44.2882    44.1726  -1.003  0.31605
## X61         -100.0564    51.6616  -1.937  0.05277 .
## X62            0.4854    51.0459   0.010  0.99241
## X63           11.1321    44.3045   0.251  0.80161
```

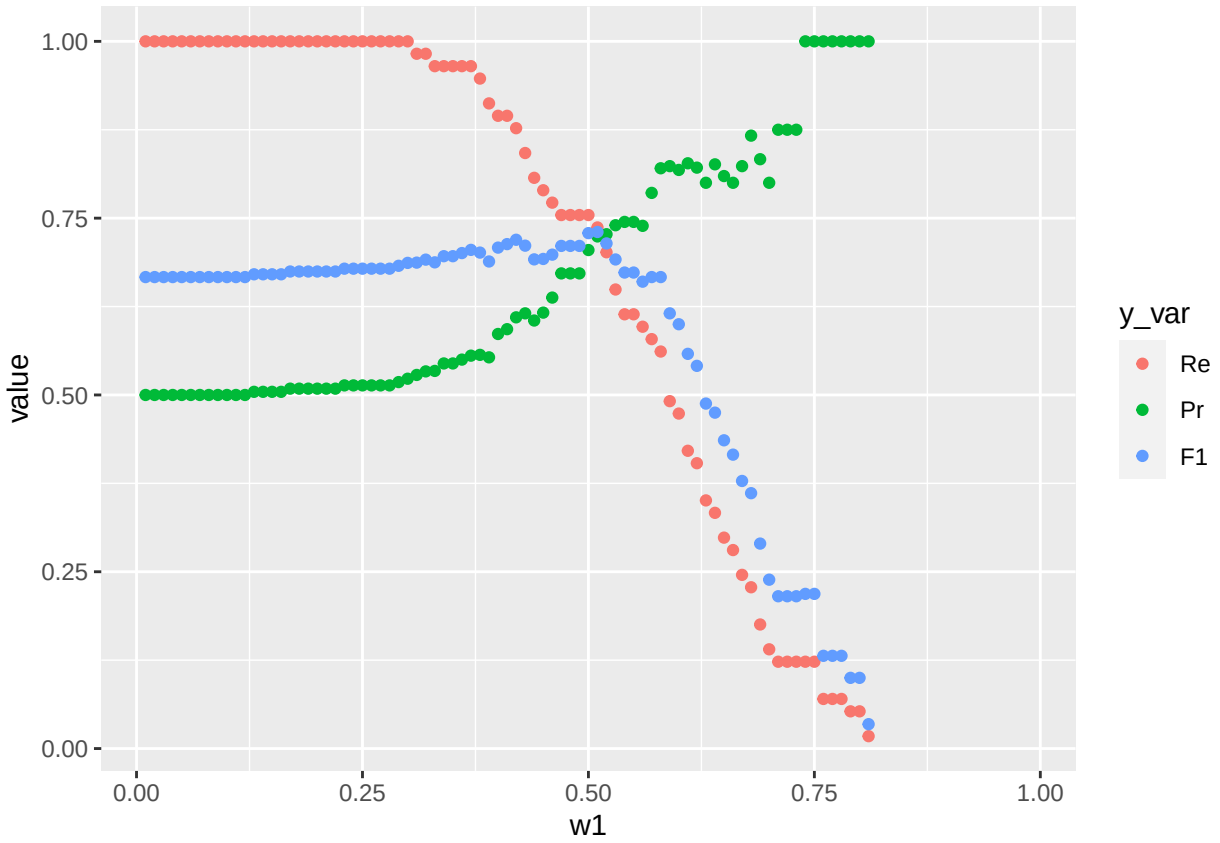
```
## X64          -21.2502    43.2332  -0.492  0.62306
## X65           15.0045    39.7810   0.377  0.70604
## X66          -13.5736    46.9650  -0.289  0.77257
## X67           82.8075    56.7168   1.460  0.14429
## X68          -32.0713    42.7246  -0.751  0.45286
## X69           23.2232    36.0940   0.643  0.51996
## X70           17.8946    44.9656   0.398  0.69066
## X71           -5.1137    43.1373  -0.119  0.90564
## X72          -38.1579    46.7366  -0.816  0.41425
## X73          -19.3768    46.4590  -0.417  0.67662
## X74          125.8558    49.8235   2.526  0.01154 *
## X75           63.5601    55.5547   1.144  0.25258
## X76           34.5685    41.9885   0.823  0.41035
## X77          -56.3144    51.3634  -1.096  0.27291
## X78           43.3761    38.4388   1.128  0.25913
## X79           15.3872    38.0375   0.405  0.68583
## X80           17.2325    49.3509   0.349  0.72695
## X81          -28.2685    50.4519  -0.560  0.57527
## X82           47.6465    38.7987   1.228  0.21943
## X83           -5.3508    46.2120  -0.116  0.90782
## X84          -25.9076    40.0323  -0.647  0.51752
## X85          -50.2329    43.7079  -1.149  0.25044
## X86           34.8850    46.0354   0.758  0.44858
## X87           31.2100    44.0543   0.708  0.47867
## X88           15.8175    38.4065   0.412  0.68045
## X89           25.0158    47.8627   0.523  0.60121
## X90           -5.5318    34.0405  -0.163  0.87091
## X91          -43.1781    36.9412  -1.169  0.24247
## X92           83.9869    40.5949   2.069  0.03856 *
## X93          -42.3146    53.8676  -0.786  0.43214
## X94            5.1914    40.4357   0.128  0.89784
## X95          -78.8831    47.3753  -1.665  0.09590 .
## X96          -12.1060    39.9755  -0.303  0.76201
## X97            8.1296    40.8431   0.199  0.84223
## X98          -39.0481    43.0483  -0.907  0.36437
## X99           58.9201    43.8725   1.343  0.17928
## X100          94.3080    45.7306   2.062  0.03918 *
## X101          39.8180    48.0345   0.829  0.40713
## X102         -68.2650    41.7882  -1.634  0.10234
## X103         -35.6032    39.4645  -0.902  0.36697
## X104         -44.1205    42.0108  -1.050  0.29362
## X105        -101.8904    60.9158  -1.673  0.09440 .
## X106          95.2479    41.6531   2.287  0.02221 *
## X107         -52.0621    39.7229  -1.311  0.18998
## X108          -5.4900    43.3830  -0.127  0.89930
## X109         -20.6792    47.9246  -0.431  0.66611
## X110         -23.3287    41.3586  -0.564  0.57271
## X111         -42.2980    42.1524  -1.003  0.31564
## X112         -53.0664    53.7175  -0.988  0.32321
## X113         -21.0303    45.0524  -0.467  0.64065
## X114         -28.4402    41.2435  -0.690  0.49047
## X115         -90.8802    42.3774  -2.145  0.03199 *
## X116          13.5827    39.0609   0.348  0.72804
## X117         -11.7948    45.0958  -0.262  0.79367
```

```
## X118          -95.6250    52.2519  -1.830  0.06724 .
## X119           -0.8448    38.9932  -0.022  0.98271
## X120          -31.2280    51.4784  -0.607  0.54410
## X121           70.4017    45.9560   1.532  0.12554
## X122          -30.0535    47.0638  -0.639  0.52310
## X123          -43.9208    41.0033  -1.071  0.28410
## X124            2.5998    57.1726   0.045  0.96373
## X125          -17.4598    43.8459  -0.398  0.69048
## X126           -8.4158    58.5711  -0.144  0.88575
## X127          -99.1520    50.5864  -1.960  0.04999 *
## X128           21.3206    44.2029   0.482  0.62957
## X129          -54.6896    39.7913  -1.374  0.16931
## X130          -11.8780    48.4810  -0.245  0.80645
## X131          -43.3556    43.0770  -1.006  0.31419
## X132          -66.7408    58.2839  -1.145  0.25217
## X133           26.9711    43.9731   0.613  0.53964
## X134          -95.9037    92.8438  -1.033  0.30162
## X135            2.7952    41.9692   0.067  0.94690
## X136           26.4698    42.7226   0.620  0.53554
## X137           84.8894    49.2221   1.725  0.08460 .
## X138           43.7769    56.4836   0.775  0.43832
## X139         -103.2298    45.5995  -2.264  0.02358 *
## X140           31.7883    54.9734   0.578  0.56310
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 315.81  on 467  degrees of freedom
## Residual deviance: 152.72  on 327  degrees of freedom
## AIC: 404.51
##
## Number of Fisher Scoring iterations: 7
```




```
##
## Call:
## glm(formula = y ~ ., family = binomial, data = train_data, weights = weight_vec)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.57109  -0.62563   0.09708   0.48343   1.43248
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.0303     0.1956   5.269 1.37e-07 ***
## PC1           6.4047     2.8986   2.210  0.02714 *
## PC2          -3.8279     5.0606  -0.756  0.44940
## PC3          13.2467     4.9990   2.650  0.00805 **
## PC4           8.2091     5.7883   1.418  0.15612
## PC5          -6.0659     7.9331  -0.765  0.44449
## PC6         -14.1471     8.1729  -1.731  0.08345 .
## PC7          11.2033     7.3219   1.530  0.12599
## PC8          12.4776     7.6152   1.639  0.10132
## PC9           3.9134     6.7969   0.576  0.56478
## PC10         -0.7104     6.8882  -0.103  0.91786
## PC11          3.2063     7.9963   0.401  0.68844
## PC12         -6.0282     7.4159  -0.813  0.41629
## PC13          2.9744     7.6349   0.390  0.69685
## PC14         -6.8906     9.5441  -0.722  0.47031
## PC15          0.8098     6.8888   0.118  0.90642
```

```
## PC16         16.8323     8.0630   2.088  0.03683 *
## PC17          2.6973     7.7466   0.348  0.72770
## PC18          0.9800     8.0336   0.122  0.90291
## PC19         -4.1053     7.9354  -0.517  0.60492
## PC20        -24.1668     9.0175  -2.680  0.00736 **
## PC21          8.2134     8.0710   1.018  0.30885
## PC22          2.9331     8.8559   0.331  0.74049
## PC23          3.7671     9.2328   0.408  0.68326
## PC24         18.2671    10.0369   1.820  0.06876 .
## PC25         -2.0375     8.7116  -0.234  0.81507
## PC26         11.5808     9.8797   1.172  0.24113
## PC27          9.9338     8.8320   1.125  0.26069
## PC28          7.3883    10.4627   0.706  0.48009
## PC29        -26.6001    10.6909  -2.488  0.01284 *
## PC30        -18.8279     9.9668  -1.889  0.05888 .
## PC31          2.8860    10.6768   0.270  0.78693
## PC32          1.0680     9.9099   0.108  0.91418
## PC33         -1.6641    10.5334  -0.158  0.87447
## PC34          1.4255    11.1711   0.128  0.89846
## PC35         -3.2596    11.9300  -0.273  0.78468
## PC36         -0.2490    10.3020  -0.024  0.98071
## PC37         13.0444    11.3278   1.152  0.24951
## PC38         -2.6376    11.3897  -0.232  0.81686
## PC39        -11.4820    12.1286  -0.947  0.34380
## PC40         13.9751    11.2222   1.245  0.21302
## PC41         -7.5064    12.3284  -0.609  0.54261
## PC42         -1.9037    11.9833  -0.159  0.87377
## PC43         11.3650    12.7197   0.893  0.37159
## PC44         -2.8203    12.2728  -0.230  0.81825
## PC45         31.8603    13.4366   2.371  0.01773 *
## PC46          6.4864    13.7484   0.472  0.63708
## PC47         -9.6011    13.3579  -0.719  0.47229
## PC48         22.2676    13.2394   1.682  0.09259 .
## PC49         11.5070    14.5456   0.791  0.42889
## PC50        -45.5011    14.7374  -3.087  0.00202 **
## PC51         24.1219    14.5953   1.653  0.09839 .
## PC52          9.0402    13.6102   0.664  0.50655
## PC53        -10.7013    14.6149  -0.732  0.46403
## PC54        -21.4388    15.3362  -1.398  0.16214
## PC55         -6.0535    14.7755  -0.410  0.68203
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 275.66  on 467  degrees of freedom
## Residual deviance: 205.71  on 412  degrees of freedom
## AIC: 226.18
##
## Number of Fisher Scoring iterations: 6
```




```
##
## Call:  glmnet(x = x_train, y = as.matrix(training_fit[, 1]), family = "binomial",      weights = wei
##
##   Df  %Dev  Lambda
## 1 32 16.12 0.02363
```

Introduction

Reddit is an American social news website that hosts discussion boards where users can share, comment and vote on various posts (Reddit wikipedia). These posts are housed in subreddits which are communities on Reddit focused on a specific topic.

When writing comments on Reddit, users will often write /s at the end of their post to indicate their comment is Sarcastic. This, coupled with Reddit's web scrapping Python API, provides a self labeled data set of sarcastic comments.

The goal our analysis will be to use the /s as a binary indicator of a comment being sarcastic and fit a Logistic regression model using various feature extraction methods. We can then explore this model's efficacy and optimize it for prediction.

Data Collection Method

On the subreddit dataisbeautiful one user posted the following figure (figure citation):

We began by scrapping the top 10,000 posts from each of the above subreddits. We found that all the subreddits had approximately a 1:100 ratio for sarcastic to non-sarcastic comments. We constructed our first data set by sampling from all the above subreddits however, we found the data to be too 0 heavy and no

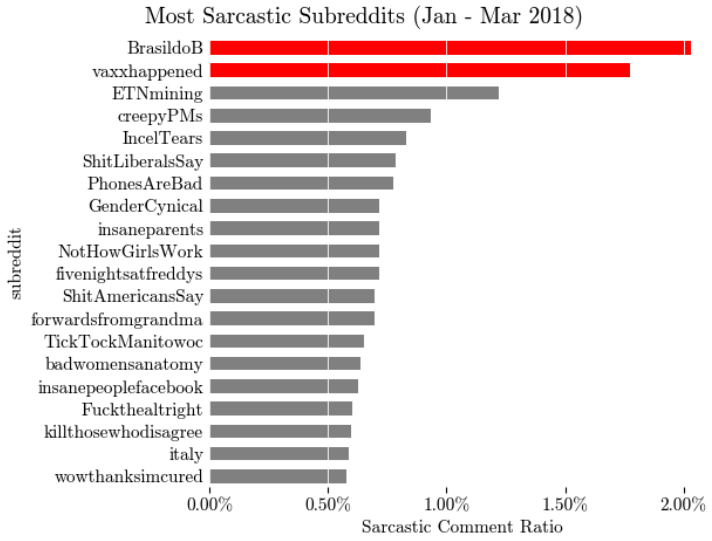


Figure 1: sarcastic_subreddits

model specification could learn an underlying relationship between words and sarcasm. We then attempted to fit models to various ratios of sarcastic to non-sarcastic comments. We found that Logistic regression began to perform reasonably well at a ratio of 1:1 sarcastic to non-sarcastic. We also found that models tended to perform far better if all comments came from a single subreddit as opposed to multiple. As per our prelimnary results we opted for a single subreddit at a ratio of 1:1 sarcastic to non sarcastic comments. NotHowGirlsWork was found to have the largest count of Sarcastic comments at 321 therefore we selected this subreddit for our data set.

Variable Description

Our data set is constructed as follows:

Variable Name	Data Type
Body	String
Sarcastic	Binary Integer

Figure 2: data set table

where Body is the raw comment string and Sarcastic is 1 when /s is present in the comment and 0 when it is not.

Data Preprocessing

In Natural Language Processing there are various text preprocessing steps that are common to employ (text as data citation):

- Punctuation, whitespace, and number removal - any punctuation characters such as !, @, #, etc. as well as empty space and numbers are removed.
- Stopword Removal - removal of words that fail to provide much contextual information, e.g., articles such as 'a' or 'the'.
- Stemming - identifying roots in *tokens*, individual words, and truncating them to their root, e.g., fishing and fisher transformed to fish.

In our data set we first removed the /s from every sarcastic comment and preformed the above preprocessing steps.

Feature Extraction Methods

In order to use text as data in a Logistic regression we must numerically encode our strings. There are a plethora of feature extraction methods in NLP. For our analysis we compare TF-IDF, Word2Vec, and GloVe.

TF-IDF

Term frequency inverse document frequency (TFIDF) is a heuristic to identify term importance (text mining in R citation). It calculate the frequency with which a term appears and adjusts it for its rarity. Rare terms are given increased values and common terms are given decreased values (text as data citation).

TFIDF is given by

$$\text{TFIDF}(t) = \text{TF}(t) \times \text{IDF}(t)$$

where

$$\text{TF}(t) = \frac{\text{\# of times term t appears in a document}}{\text{\# of terms in the document}}$$

and

$$\text{IDF}(t) = \ln\left(\frac{\text{\# total number of documents}}{\text{\# number of documents where t appears}}\right)$$

In our analysis a document is a Reddit comment. After being preprocessed, the text of each comment is separated into tokens and has its TFIDF calculated. From there the TFIDF values are placed in a *Document Term Matrix* (DTM). This matrix has document ids as rows and tokens as columns. It is therefore a sparse matrix where entries are the TFIDF scores for corresponding tokens.

The DTM acts as the design matrix for our Logistic Regression model:

```
## <<DocumentTermMatrix (documents: 6, terms: 8)>>
## Non-/sparse entries: 1/47
## Sparsity           : 98%
## Maximal term length: 10
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
## Sample             :
##     Terms
## Docs common forevaaaaa husband lost potenti surviv two      will
##   10      0          0       0    0       0      0   0 0.0000000
##   5       0          0       0    0       0      0   0 0.0000000
##   6       0          0       0    0       0      0   0 0.2352558
##   7       0          0       0    0       0      0   0 0.0000000
##   8       0          0       0    0       0      0   0 0.0000000
##   9       0          0       0    0       0      0   0 0.0000000
```

Word2Vec

Word2Vec is a group of predictive models for learning vector representations of words from raw text. Word2Vec uses either the *continuous Bag-of-Words architecture* (CBOW) or the *continuous Skip-Gram architecture* (Skip-Gram) to compute the continuous vector representation of words. Both CBOW and Skip-Gram use shallow neural networks to achieve this, but CBOW predicts words based on the context and Skip-Gram predicts surrounding words given the current word (Efficient Estimation of Word Representations in Vector Space paper citation).

Each word is represented as a vector, and words that share common context are close together in vector space (Deep Learning Essentials textbook citation). Document vectors are representations of documents (Reddit comments) in vector space. A document vector can be constructed by summing the the word vectors from a common document and then standardizing them (word2Vec package citation). The design matrix for logistic regression can be constructed with the rows of the matrix as the document vectors. The resulting design matrix therefore has one row per Reddit comment and is as follows:

GloVe

Global vectors for word representation (GloVe) is an unsupervised learning algorithm which creates a vector representation for words by aggregating word co-occurrences from a corpus. The resulting co-occurrence matrix X contains elements X_ij representing how often word i appears in the context of word j (citation).

Next, soft constraints for each word pair are defined by:

$$w_i^T w_j + b_i + b_j = log(X_{ij})$$

where w_i is the vector for the main word, w_j is the vector for the context word j, and b_i and b_j are scalar biases for the main and context words. Finally, a cost function is defined:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - \log\, X_{ij})^2$$

Here f is a weighting function chosen by the GloVe authors to prevent solely learning on extremely common word pairs (citation):

$$f(X_{ij}) = \begin{cases} \left(\frac{X_{ij}}{x_max}\right)\alpha & \text{if } X_{ij} < XMAX \\ 1 & \text{otherwise} \end{cases}$$

To create the design matrix below, a vocabulary of the words in the corpus was created. Since this method creates a co-occurrence matrix, we prune all words which appear less than five times to reduce bias from less common words (citation). From there we constructed a term-co-occurrence matrix and factorized it via the GloVe algorithm. The resulting matrix consists of word vectors as rows, which are added together to create sentence vectors that are used to train the model:

Evaluation Metrics

Model performance is assessed based on classification performance. In sentiment analysis the most common metrics to tune model for performance are Precision, Recall, and F1 Score (citation).

Precision is the number of true positive divided by the number of true and false positives. Recall is the number of true positive divided by false negatives and true positive. It is the true positive rate. F1 Score is the harmonic mean of Recall and Precision (python learning citation) (add a CM and write the formulas).

For our analysis a true positive is a correctly predicting a comment is sarcastic.

Regression Analysis

This analysis is a comparison of logistic regression model performance when using 3 types of feature extraction. For each feature extraction we fit a base model. We then perform Principal Componenet Analysis (PCA) to reduced dimensionality and deal with multicollinearity. Finally we investigate LASSO models a means for dimensionality reduction.

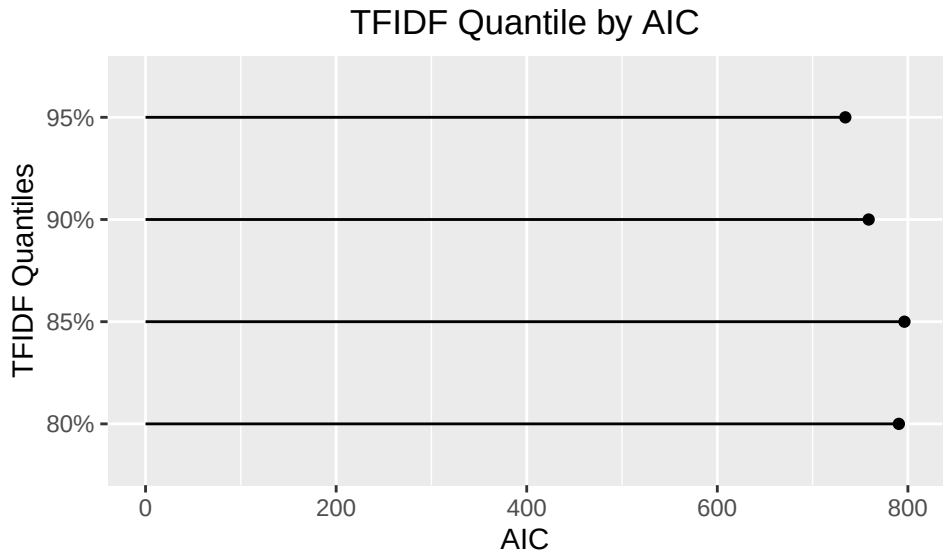
After model fitting we perform weighting on all 3 model types and decide a best model for each feature extraction method. Weighting in done by multiply each predictor by w, where $w \in (0, 1)$, if a comment is sarcastic and by $1 - w$ if a comment is not sarcastic. This is done for every w starting from 0.01 to 0.99 in increments of 0.01. We record testing and training metrics foe each model and discuss our optimal selection.

We hypothesis that models using GloVe as the feature extraction method will perform similarity to Word2Vec. TF-IDF will perform the worst. TF-IDF numerically encodes text based on rarity. We think this is too simple an approach to capture important sarcastic words. Word2Vec captures context and GloVe interprets word co-occurrences which we believe could both be suitable strategies to capture sarcastic structure in text.

Variable Selection

TF-IDF

After performing text preprocessing and TFIDF calculations the resulting DTM was 642×2074. This matrix has far too many columns compared to rows and so some dimensionality reduction was required. One way to do so is to filter away unimportant terms. This can be decided by the percentiles of the TF-IDFs. For a given percentile we can exclude columns of the DTM based on whether or not their values fall within that percentile. We do this in increments of 5 from the 5th percentile to the 95th percentile, fit a model, and recording the corresponding AIC. We opt to keep the DTM that produced the model with the lowest AIC.



The model corresponding to the lowest AIC had a DTM filtered to disclude TFIDF values below the 95th percentile resulting in a 512×74 matrix.

Word2Vec

GloVe

Fitting, Evaluations, and Violations

fit base

show multicollinearity

do pca

discuss LASSO as another option instead of PCA

TF-IDF

For the base model we take the minimum AIC model found during variable selection. This result model has a 14% reduction in deviance and an AIC of 734. R fails to estimate several predictor coefficients and outputs them of NA. This suggests investigation of multicollinearity is needed. After examining the model's VIFs and the correlation between predictors it is was found that 6 variables have VIfs that are over 10 and several predictors are perfectly correlated. To deal with multicollinearity PCA and LASSO are explored.

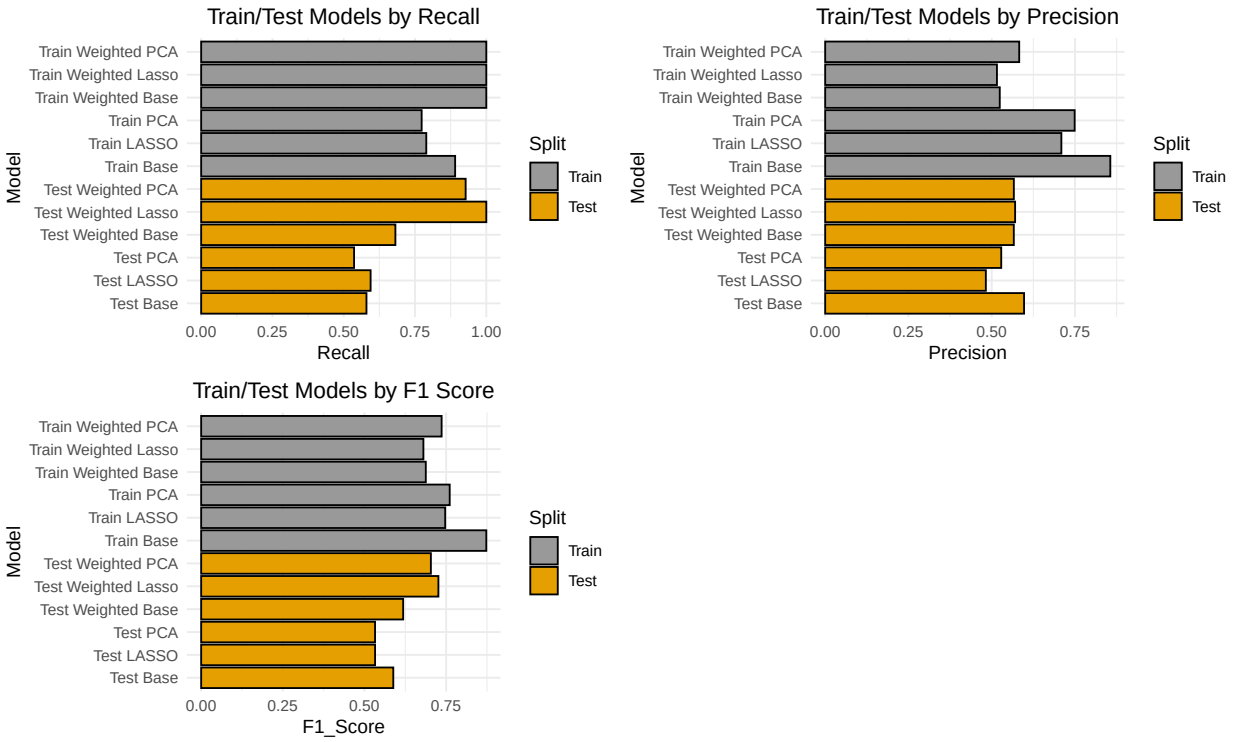
For PCA we kept a cumulative proportion of up to 90% which resulted in using 24 principal components. Fitting our model to the data the AIC was 17135 and the deviance increase by a factor of 24. Clearly the model does not fit the data well. However, no VIFs were found to be over 10 so the multicollinearity was removed.

Next we employed cross validation to fit a LASSO model. An optimal λ of 0.025 was selected and the resulting model produced a 34.55% reduction in deviance.

We know move onto optimal weight selection. We seek to achieve the best balance of Precision, Recall, and F1 Score.

Examining the above graphs the best weightings are 0.75 for the base model, 0.62 for the PCA modek, and 0.55 for the LASSO model.

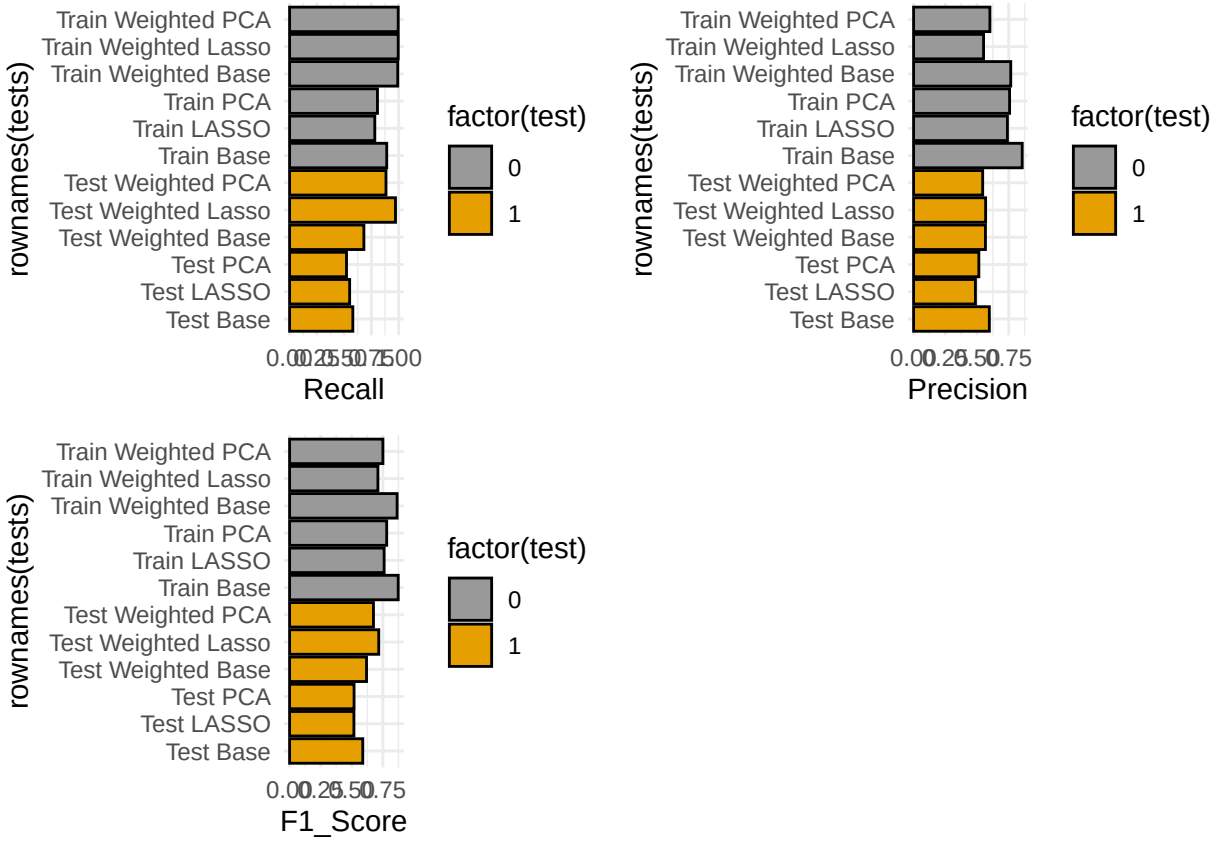
With model selection finished we may now compare all models and pick the best TFIDF model.

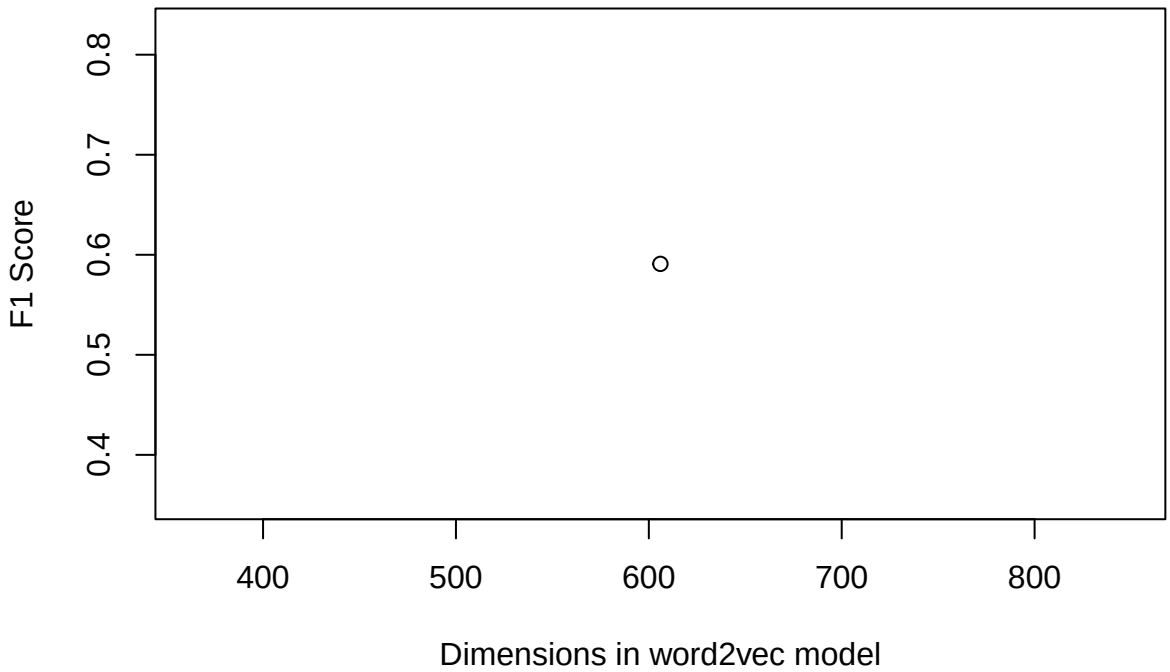


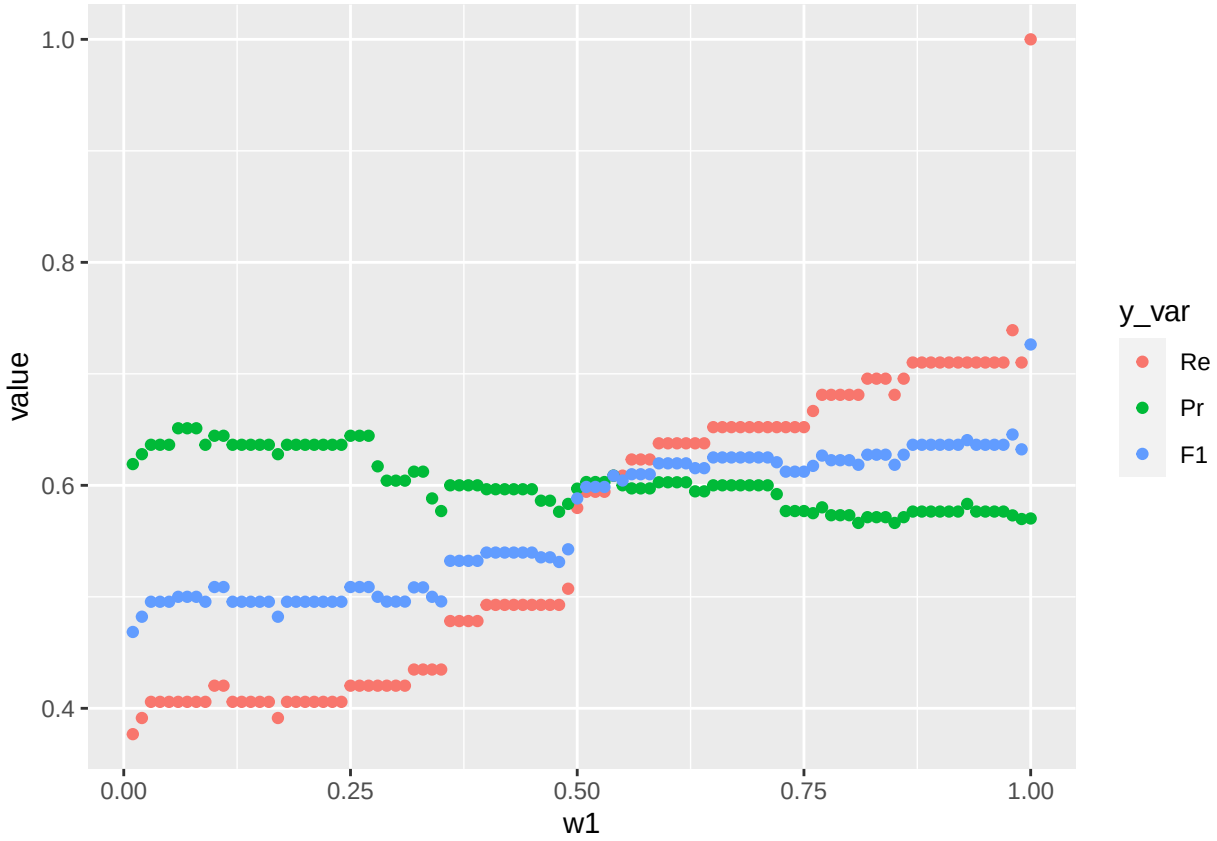
Weighted LASSO is the superior model. While it has lower Recall than Weighted PCA and Base, it does the best in F1 Score which indicates it is the most balanced model. Both Weighted PCA and Weighted base have a poor Precision and F1 Scores

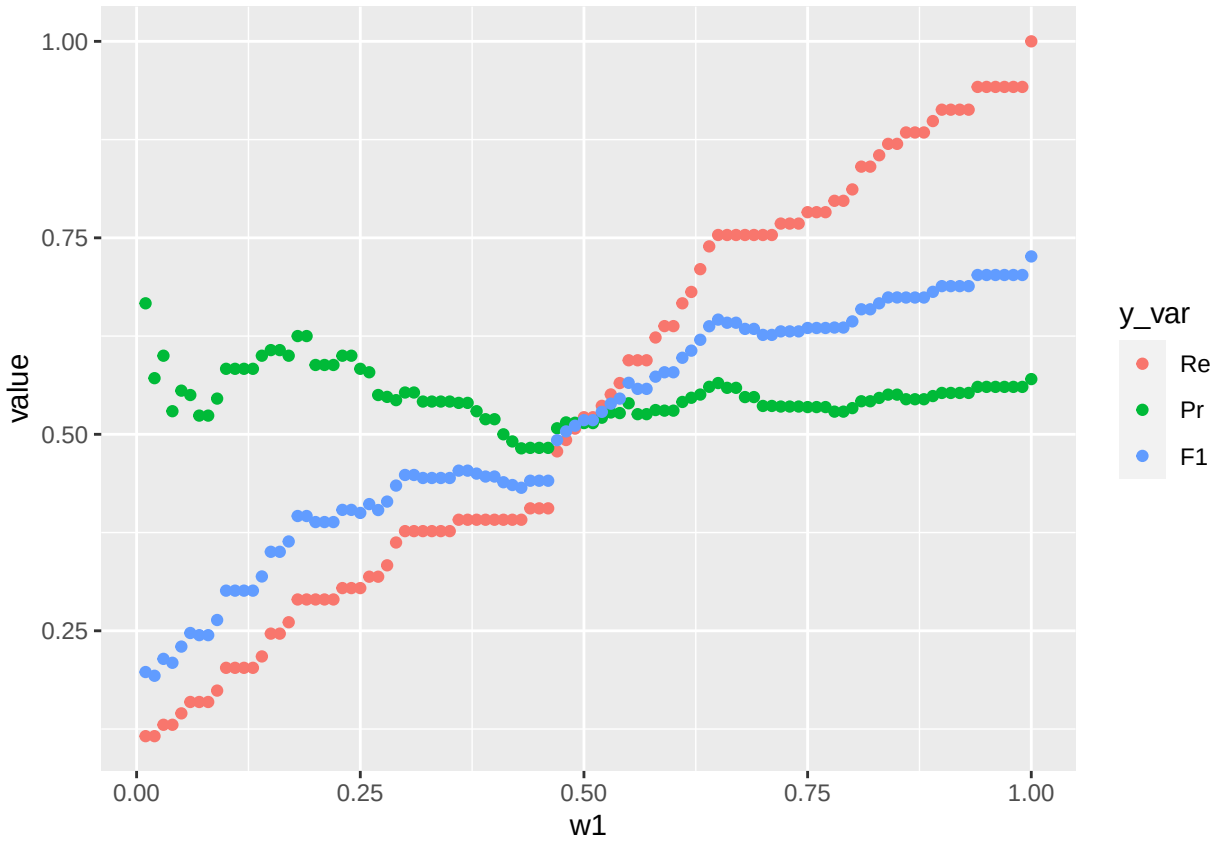
Word2Vec

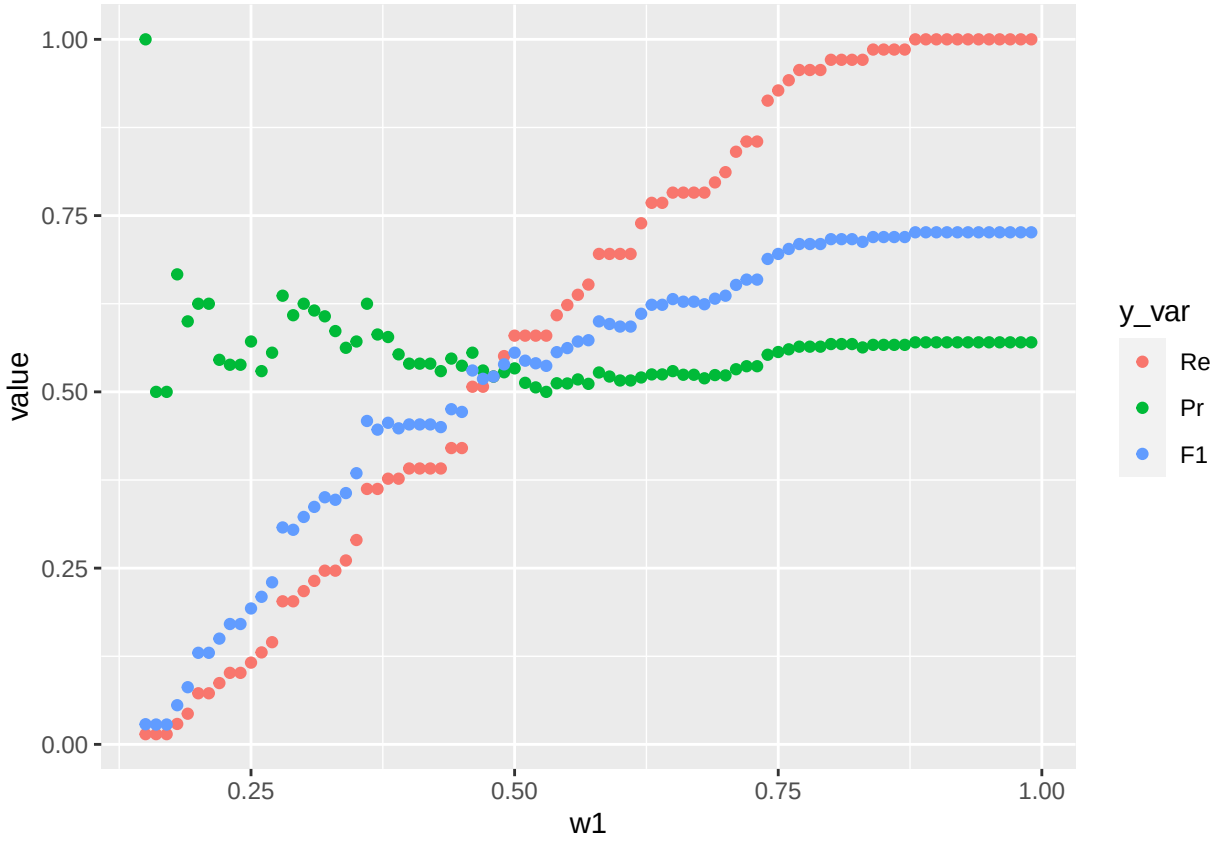
GloVe






1.0
0.8
0.6
0.4
value
0.00
0.25
0.50
0.75
1.00
w1
y_var
Re
Pr
F1


1.00
0.75
0.50
0.25
value
0.00
0.25
0.50
0.75
1.00
w1
y_var
Re
Pr
F1



Other Findings

TF-IDF

As weighted LASSO was found to be the best TF-IDF model we have access to the set of words that were not shrunk to 0. This provides us we insight as to which words predict best for sarcasm in the subreddit NHGW. The words are:

tate	written	companionship	reduct	final
remov	report	nowaday	gold	realis
asexu	sister	dump	iron	page
pictur	puberti	gal	act	broke
ignor	crime	cop	polic	pedo
jail	depend			

The results are quite interesting. NHGW is a subreddit about making fun of those who seemingly unaware of why women act the way they do and we see terms related to sexuality, relationships, and crime. Notably we also see *tate* a highly controversial figure for his views on women.

Word2Vec

GloVe

Conclusion

Model Comparison

of all methods which had the best balance of metrics

Limitations

The largest issue with this analysis is the dataset. Logistic regression is not equipped to handle such 0 heavy data and without us artifically cosntructing the ratio of sarcastic to nonsarc comments this analysis likely would not work.

discainer about data set and stocahstic nature of Glove and Word2Vec

Final Remarks

Summary of everything

References

https://en.wikipedia.org/wiki/Reddit#References

https://www.reddit.com/r/dataisbeautiful/comments/9q7meu/most_sarcastic_subreddits_oc/

Text as Data Barry DeVille, Gurpreet Singh Bawa

This paper shows we can use these metrics for this: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9320958&tag=1

for definition of recall, precision, and f1 use: Hands-On Ensemble Learning with Python