Hemtenta

950114-9638 Agaton Fransson

June 2020

1 Exercise 1

OBS: I write "we" throughout the exam but I did everything alone. I also chose to demonstrate the results of my exploratory data analysis in the methods since the results of the exploration justifies the other methods.

1.1 Methods

1.1.1 Exploratory data analysis with results

The first thing we notice about the training data is that we have 800 dimensions and 323 samples, more dimensions than samples. Hopefully not all features are relevant to classify the data points so that we can reduce the dimensionality of it. To visualize the data of the two classes before removing any dimensions we can make stem plots (using the python's library matplotlib.pyplot.stem) of the mean of each dimension within the two classes.

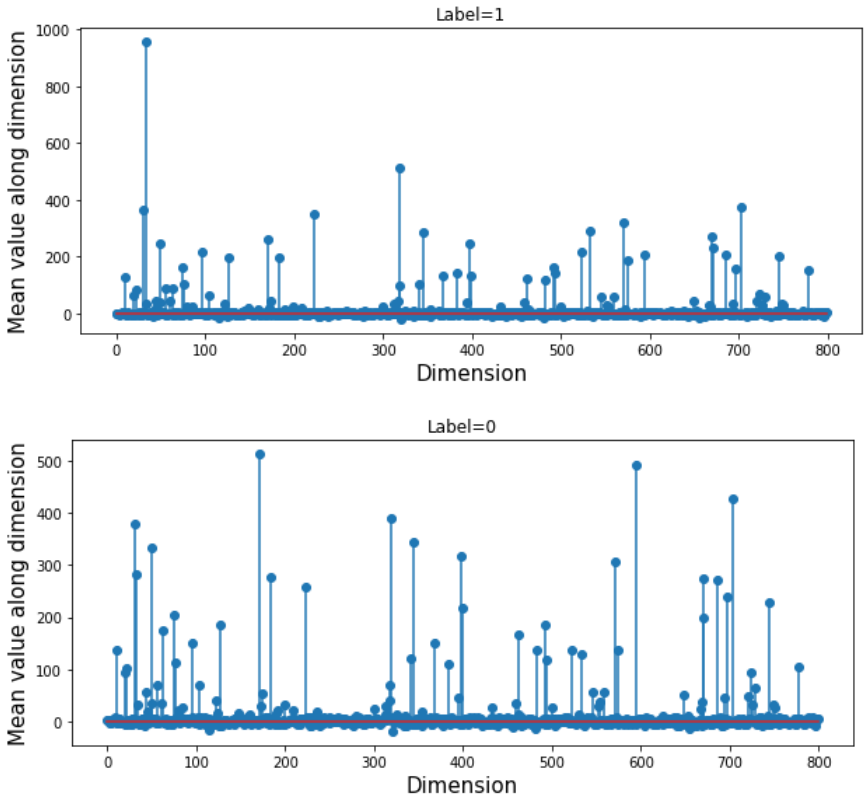

Figure 1: Stem plots for the mean value of the data of the training set along each dimension. The top plot is for class 1 and the bottom one is for class 0. We see that the majority of the dimensions seem to be noise.

The plots in figure (1) suggests that we have a sparse data set and that only a few features out of the 800 are important to actually classifying the two classes. We also notice that the means don't have any large negative values. We see that class 1 has 76 data points while class 0 has 247 data points out of 323 for the training set, and 41 data points in class 1 and 134 data points in class 0 out of 175 data points for the validation set. A method that reduces the dimensionality of our data set should be worth a try.

1.1.2 ℓ_1-regulated logistic regression

To reduce the dimensionality of the data we first try the ℓ_1-regulated logistic regression (aka lasso), using sklearn.linear_model.LogisticRegression in python. The lasso regression, equation from lectures, performs a feature selection through minimizing

$$\hat{\boldsymbol{\beta}}_{lasso}(\lambda) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \, ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1$$

where $\mathbf{X}$ is the data set, $\mathbf{y}$ is the targets, $\boldsymbol{\beta}$ is the solution, λ is the regularization strength (the parameter that decides how many features to remove basically, it penalizes a $\boldsymbol{\beta}$ with many features) and $\hat{\boldsymbol{\beta}}_{lasso}$ is the solution with, hopefully, fewer features. It is obvious that a high λ increases the penalty from a $\boldsymbol{\beta}$ with many elements and high values since this makes the norm of $\boldsymbol{\beta}$ higher. We fit the model to the training data and then compute the score of it testing on the validation data. We do this for several values of the hyperparameter $C = \frac{1}{\lambda}$ (inverse of regulatization strength) to find the optimal value. This generates the plot seen in figure 2 in the results. In this plot we also see how well the classifier could classify data of the two different classes separately.

The resulting coefficients in $\hat{\boldsymbol{\beta}}_{lasso}$ are then visualized in figure 3 and their indices listed in table 1. To see that this solution isn't complete nonsense we can multiply the coefficients with the mean of each feature for the two classes and compute the sum to see if the answer is close to the label. These sums are seen along with the stem plots for the two class means multiplied with the resulting coefficients in figure 4 in the results.

1.1.3 Shrunken centroids

The second method we try is the shrunken centroids, using sklearn.neighbors.-NearestCentroid in python. The shrunken centroids is a variation of the nearest centroid method which assigns data points to their nearest class centroid. The shrunken centroids algorithm introduces a threshold that sets features of a class centroid to equal the mean of all data along that feature if they are close enough to it. This means that we decrease the number of features that decides how a data point is classified, which is what we want to do. We train the model on the training data and test it on the validation data for different values of the hyperparameter. The scores of this procedure is seen in figure 5 in the results, together with the scores of the model while classifying data from each class alone. The similarities between the two centroids with regards to nonzero features is calculated using python's function set(a).intersection(b) and is presented in the results.

1.1.4 Comparison of classifiers

When comparing the two classifiers their scores on the validation set is compared, both total score and score on classifying the two different labels. The number of features and which features they chose will be compared as well to see if there is an overlap.

1.2 Results

1.2.1 ℓ_1-regulated logistic regression

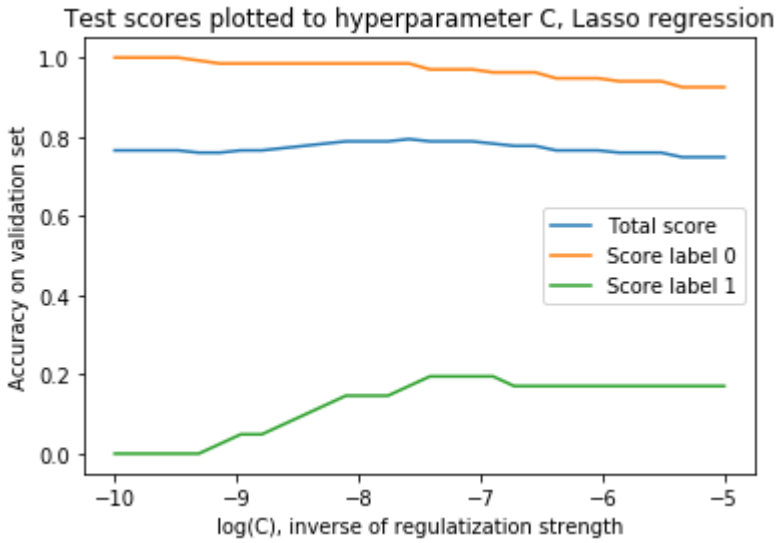

Figure 2: Accuracy of trained model on validation set plotted against the log(C) where C is the inverse of the regularization strength and log() is the natural logarithm. Accuracy of the model on the two separate labels is also shown.

Through figure 2 we find that the optimal value for $log(C) \approx -7.6$ which gives us a score of 0.79 on the validation set, where classification of label 0 had an accuracy of 0.98 och label 1 had 0.17. 18 of the original 800 features remain and they are listed in table 1 and their values are seen in figure 3.

Remaining features					
10	31	33	63	171	183
319	341	399	462	483	492
533	571	594	686	703	745

Table 1: List of remaining features after ℓ_1-regulated logistic regression.

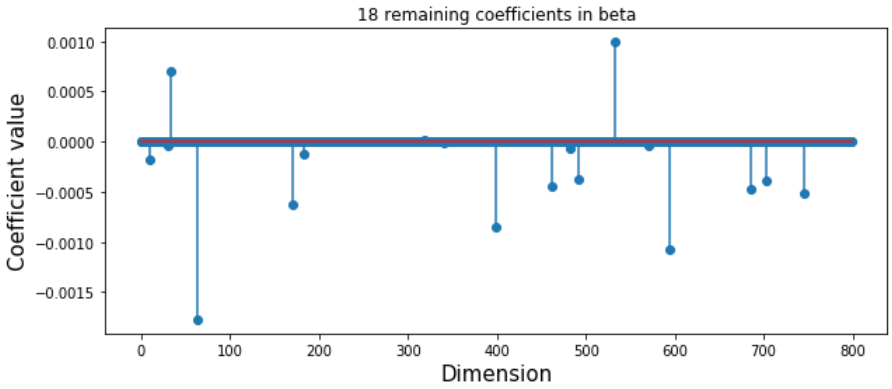

Figure 3: Remaining features after ℓ_1-regulated logistic regression with the values of their coefficients on the y-axis.

We see from figure 3 that some features will have an almost negligible impact when computing $\boldsymbol{\beta X}$ since their amplitude is so small. We also see that most of them are negative. The means along each feature for the two classes in the validation set multiplied by the resulting $\hat{\boldsymbol{\beta}}_{lasso}$ from the model is presented in figure 4 along with the resulting sums in the titles.

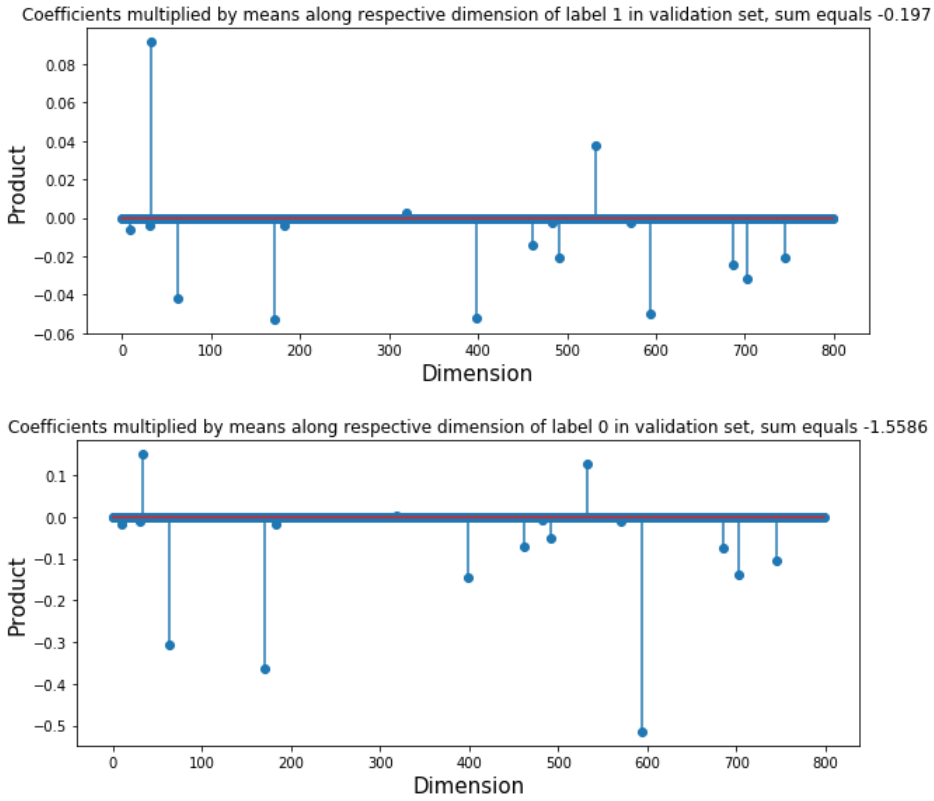

Figure 4: Stem plots over the resulting coefficients multiplied by the means along each dimension of the validation set for the two labels. Resulting sum written in the plot titles.

In the titles of figure 4 we see that the sums are both negative, but the sum for class 0 is lower than the sum for class 1. This can also be seen in the figures where the positive terms have higher absolute values in the plot for label 1 compared to the plot of label 0, and the negative terms have larger absolute values in the plot of label 0 compared to the plot of label 1.

1.2.2 Shrunken centroids

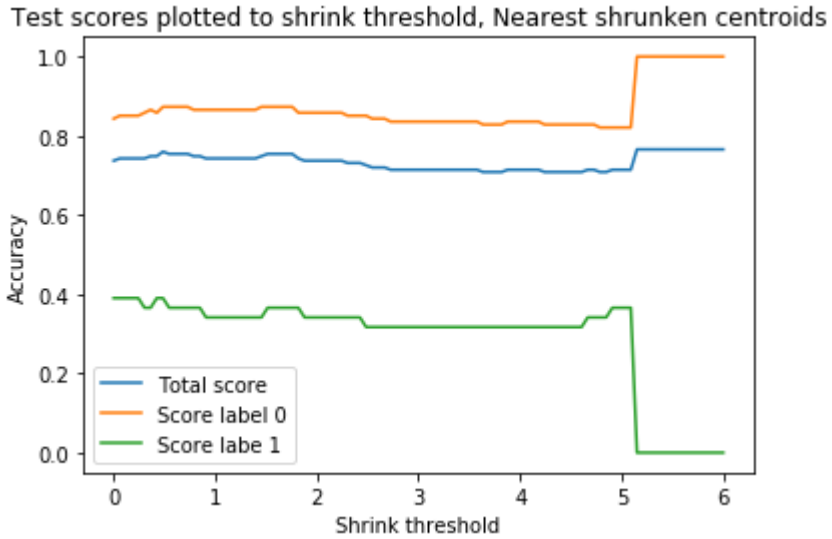

Figure 5: Accuracy of trained model NearestCentroids on validation set plotted against the shrink threshold. Accuracy of the model on the two separate labels is also shown.

We see that the highest score possible is achieved when the shrink threshold is higher than ≈ 5.2. This gives an accuracy of ≈ 0.766, with 0.0 accuracy on label 1 and 1.0 accuracy for label 0. For this threshold the two resulting centroids had the same 800 coefficients. The optimal shrink threshold lower than 5 was ≈ 0.46 and generated a total accuracy of 0.76, and got 0.87 on label 0 and 0.39 on label 0. The two centroids had 457 coefficients in common for this value, the features that differed are seen in figure 6.

```
Features that had different coefficients in the two centroids:
[  0   1   2   3   7  13  17  18  19  20  22  23  24  29  30  32  33  37
  38  39  41  44  45  50  51  52  54  55  56  58  59  60  62  63  66  67
  68  70  71  73  74  75  79  81  85  86  92  96  97  98  99 100 107 108
 110 112 114 121 122 123 124 126 132 133 134 135 137 139 142 144 146 147
 151 155 156 162 163 167 169 171 172 174 177 179 180 182 183 184 185 186
 187 189 191 192 196 197 199 200 201 211 212 213 214 215 216 220 222 223
 228 230 232 235 236 240 244 245 255 256 257 259 262 266 269 272 276 277
 280 281 285 287 288 289 290 291 296 298 310 311 312 313 314 317 318 319
 322 323 328 329 332 333 335 336 339 342 344 345 346 347 348 349 353 355
 356 360 363 365 368 378 381 383 391 393 394 396 398 399 401 402 404 405
 406 407 408 410 416 419 420 422 429 430 432 434 436 437 440 442 447 449
 452 454 458 460 462 463 464 468 470 471 474 476 478 481 482 483 488 492
 494 496 502 506 511 513 514 515 521 522 523 524 528 533 539 542 548 549
 550 553 554 557 561 562 563 567 568 573 575 578 579 582 583 586 587 588
 589 590 593 594 596 598 599 600 608 609 612 616 618 619 629 630 636 638
 639 640 643 644 645 647 648 649 651 654 655 656 660 665 666 667 668 669
 671 674 677 679 682 683 684 685 686 688 689 691 693 694 695 696 697 700
 701 702 703 707 709 714 716 719 720 724 725 729 732 736 739 740 741 743
 745 755 756 760 761 773 774 778 783 786 787 788 789 790 793 794 796 797
 798]
```

Figure 6: The features that had different values in the two centroids.

The features in figure 6 are the ones deemed important by the model since these weren't close enough to the overall centroid to be set to the overall centroid's value.

1.2.3 Comparison of selected features

Using python's function intersection() we found that 14 features selected by ℓ_1-regulated logistic regression were present amongst the features selected by shrunken centroids for the lower value of the shrink threshold, all features except for 10, 31, 341 and 571 were mutual.

1.3 Discussion

1.3.1 High values of the shrink threshold

The shrunken centroids method generated a maximum accuracy of ≈ 0.766 which is the same accuracy we would achieve if we classified all data points in the validation set to class 0 ($\frac{134}{175} = 0.766$). This combined with the fact that the accuracy of the shrunken centroids classifier is unchanged when increasing the value for the shrink threshold in figure 5 (and the obvious one, that they share the same 800 coefficients) tells us that we now only have one centroid which isn't that interesting when trying to find the most important features.

1.3.2 Lower values of the shrink threshold

For the lower value of the shrink threshold we got a lower total accuracy but two separate centroids and a higher accuracy on label 0, this argues that a total accuracy might not be the best way to measure the performance of the classifier for an unbalanced data set such as this one. It also suggests that the most important features to classify the data set, according to this method, are features that the two centroids have different coefficients in.

1.3.3 ℓ_1-regulated logistic regression

ℓ_1-regulated logistic regression produced a total accuracy of 0.79, an accuracy of 0.98 on label 0 and an accuracy of 0.17 on label 1, the total accuracy is higher than if all data points where put into the same class. The total accuracy would make it seem as if ℓ_1-regulated logistic regression was better at classifying our data set than shrunken centroids, but when taking into account that the data is unbalanced I would argue that the shrunken centroids performed better since ℓ_1-regulated logistic regression was more biased towards classiffying a data point as class 0.

1.3.4 Most important features

To determine the most important features I would argue that the intersection of:
1. The features that had different coefficients between the two centroids,
and
2. The features selected by ℓ_1-regulated logistic regression,
are strong candidates. We can also look at the amplitude of the coefficients from the ℓ_1-solution in figure 3 and see that the missing features (10, 31, 341 and 571) between the centroids and the ℓ_1-method had very low amplitudes which further suggests that they had a low impact. One could also point out that feature 319 and feature 483 have a low amplitude in the same figure. Following this reasoning the most important features would be:

33, 63, 171, 183, 399, 462, 492, 533, 594, 686, 703 and 745.

But since shrunken centroids used over 300 features it is very likely that lasso regression removed some important features. A softer regularization than ℓ_1 that would allow us to get more important features into the solution could likely get a higher performance on the data set. But these 12 features would very likely be part of the solution of this classifier as well, so if they aren't *the* most important features they are atleast *amongst* the most important features I would say.