Comp 551 – Applied Machine Learning

Programming Assignment 3

Sheldon Benard (260618386)

# Preface

All the preprocessing files (vocab, training, validation, test) can be found in the **data** folder of the project directory. Further, for readability, table values have been truncated to 4 decimal places.

For both sentiment classifications, a 10,000-word vocabulary of the most frequent words was used. The only preprocessing on the text input files was the removal of punctuation and the lowercasing of each of the words.

Finally, all tables outline the F1-measure, as the F1-measure was the evaluation metric for the assignment.

# Sentiment Classification – Yelp

For the yelp sentiment classification, as a baseline, the performance of a uniformly random classifier and majority-class classifier are outlined on the next page (Note: the majority class is determined from the training set):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Classifier (Set) | Class 1 | Class 2 | | Class 3 | | Class 4 | Class 5 | Ave |
| Random (Test) | 0.1134 | | 0.1282 | | 0.1623 | 0.2610 | 0.2273 | 0.1785 |
| Random (Train) | 0.1286 | | 0.1244 | | 0.1619 | 0.2617 | 0.2518 | 0.1857 |
| Random (Valid) | 0.1398 | | 0.1163 | | 0.2204 | 0.2779 | 0.2420 | 0.1993 |
| Majority  (Test) | 0 | | 0 | | 0 | 0.5196 | 0 | 0.1039 |
| Majority  (Train) | 0 | | 0 | | 0 | 0.5213 | 0 | 0.1042 |
| Majority  (Valid) | 0 | | 0 | | 0 | 0.5250 | 0 | 0.1050 |

### Binary Bag-of-Words – Yelp

Bernoulli Naïve Bayes

For Bernoulli Naïve Bayes, the hyper-parameter being tuned was α, the additive (Laplace) smoothing parameter. The details of the hyper-parameter tuning are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Hyper-parameter | Range | Best Value | Notes |
| α | (0, 10] |  | α was incremented by hundredths (0.01,0.02…) |

In the table below, the F1 performance measures (test, training, validation) for Bernoulli Naïve Bayes with the best hyper-parameter values are provided:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data set | Class 1 | Class 2 | | Class 3 | | Class 4 | Class 5 | Ave |
| Test Set | 0.416 | | 0.2638 | | 0.2404 | 0.4469 | 0.5270 | 0.3788 |
| Training Set | 0.7842 | | 0.7539 | | 0.7379 | 0.7009 | 0.7093 | 0.7372 |
| Validation Set | 0.7393 | | 0.7085 | | 0.7013 | 0.6539 | 0.6174 | 0.6841 |

Linear SVM

For Linear SVM, the hyper-parameters being tuned were the loss function and the penalty parameter C of the error term. The details of the hyper-parameter tunings are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Hyper-parameter | Range | Best Value | Notes |
| Loss function, L | {hinge, squared\_hinge} |  | The squared hinge loss and L1 penalization aren’t compatible. |
| Penalty parameter, C |  | *C = 0.1* | C was incremented by 0.1, so C = 0.1, 0.2, 0.3… |
| Penalization, P | *P = L2* | *P = L2* | P = L1 wasn’t considered due to incompatibility with squared hinge |
| Dual | *Dual = True* | *Dual = True* | Less data points than features, so solve dual |

In the table below, the F1 performance measures (test, training, validation) for Linear SVM with the best hyper-parameter values are provided:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data set | Class 1 | Class 2 | | Class 3 | | Class 4 | Class 5 | Ave |
| Test Set | 0.5180 | | 0.3145 | | 0.3175 | 0.4921 | 0.5894 | 0.4463 |
| Training Set | 0.9477 | | 0.9335 | | 0.8993 | 0.8874 | 0.8949 | 0.9126 |
| Validation Set | 0.9759 | | 0.9680 | | 0.9190 | 0.8961 | 0.8927 | 0.9303 |

Decision Trees

For Decision Tree, the hyper-parameters being tuned were: *criterion, splitter, max depth, min\_sample\_leaf,* and *max\_features*. The details of the hyper-parameter tunings are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Hyper-parameter | Range | Best Value | Notes |
| Criterion, C | {gini, entropy} |  |  |
| Splitter, S | {best, random} | *S = random* |  |
| Max Depth, D | *D*  *{None,10,20,30,40*  *,50,60,70,80,90}* | *D = None* |  |
| Min Sample Leaf, M | *M*  *{1,2,3,4,5,6,7,8,9,*  *10,20,30,40,50}* | *M = 9* |  |
| Max Features, F | *F*  *{None, 3,5,7,9,*  *10,15,20}* | *F = None* |  |

In the table below, the F1 performance measures (test, training, validation) for Decision Tree with the best hyper-parameter values are provided:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data set | Class 1 | Class 2 | | Class 3 | | Class 4 | Class 5 | Ave |
| Test Set | 0.2008 | | 0.2215 | | 0.2033 | 0.4215 | 0.4523 | 0.2999 |
| Training Set | 0.5183 | | 0.4520 | | 0.4905 | 0.6910 | 0.7103 | 0.5724 |
| Validation Set | 0.5534 | | 0.4071 | | 0.5226 | 0.6973 | 0.6985 | 0.5758 |

### Comments

Each classifier’s average F1 score outperformed the baseline classifiers (Majority, random). Further, one can notice that, on average, the classifiers had an easier time classifying higher classes (4,5) rather than lower classes (especially 2,3), probably due to the finer line between class 2 and 3 vs. class 4 and 5.

On test data, Linear SVM outperformed Naïve Bayes and Decision Trees. This is probably due to Naïve Bayes’ independence assumption (as there’s most likely a dependence between the words used) and there may be coincidental regularities in the training and validation data that cause the Decision tree to over-fit and perform worse on the test set.

### Frequency Bag of Words – Yelp

Gaussian Naïve Bayes

For Gaussian Naïve Bayes, there were no hyper-parameters to tune.

In the table below, the F1 performance measures (test, training, validation) for Gaussian Naïve Bayes are provided:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data set | Class 1 | Class 2 | | Class 3 | | Class 4 | Class 5 | Ave |
| Test Set | 0.1794 | | 0.1406 | | 0.2102 | 0.3388 | 0.3769 | 0.2492 |
| Training Set | 0.7409 | | 0.7225 | | 0.7337 | 0.7920 | 0.8370 | 0.7652 |
| Validation Set | 0.7567 | | 0.7343 | | 0.6798 | 0.7275 | 0.7934 | 0.7383 |

Linear SVM

For Linear SVM, the hyper-parameters being tuned were the loss function and the penalty parameter C of the error term. The details of the hyper-parameter tunings are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Hyper-parameter | Range | Best Value | Notes |
| Loss function, L | {hinge, squared\_hinge} |  | The squared hinge loss and L1 penalization aren’t compatible. |
| Penalty parameter, C |  | *C = 0.1* | C was incremented by 0.1, so C = 0.1, 0.2, 0.3… |
| Penalization, P | *P = L2* | *P = L2* | P = L1 wasn’t considered due to incompatibility with squared hinge |
| Dual | *Dual = True* | *Dual = True* | Less data points than features, so solve dual |

In the table below, the F1 performance measures (test, training, validation) for Linear SVM with the best hyper-parameter values are provided:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data set | Class 1 | Class 2 | | Class 3 | | Class 4 | Class 5 | Ave |
| Test Set | 0.4313 | | 0.2509 | | 0.1479 | 0.4281 | 0.6209 | 0.3758 |
| Training Set | 0.6371 | | 0.5205 | | 0.4083 | 0.5037 | 0.6555 | 0.5450 |
| Validation Set | 0.6231 | | 0.6013 | | 0.4304 | 0.5038 | 0.6269 | 0.5571 |

Decision Trees

For Decision Tree, the hyper-parameters being tuned were: *criterion, splitter, max depth, min\_sample\_leafs,* and *max\_features*. The details of the hyper-parameter tunings are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Hyper-parameter | Range | Best Value | Notes |
| Criterion, C | {gini, entropy} |  |  |
| Splitter, S | {best, random} | *S = best* |  |
| Max Depth, D | *D*  *{None,10,20,30,40*  *,50,60,70,80,90}* | *D = 50* |  |
| Min Sample Leaf, M | *M*  *{1,2,3,4,5,6,7,8,9,*  *10,20,30,40,50}* | *M = 40* |  |
| Max Features, F | *F*  *{None, 3,5,7,9,*  *10,15,20}* | *F = None* |  |

In the table below, the F1 performance measures (test, training, validation) for Decision Tree with the best hyper-parameter values are provided:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data set | Class 1 | Class 2 | | Class 3 | | Class 4 | Class 5 | Ave |
| Test Set | 0.192 | | 0.1597 | | 0.1894 | 0.4361 | 0.4575 | 0.2869 |
| Training Set | 0.3692 | | 0.2585 | | 0.2784 | 0.5376 | 0.5848 | 0.4057 |
| Validation Set | 0.4444 | | 0.1805 | | 0.3 | 0.5350 | 0.6084 | 0.4136 |

### Comments

Each classifier’s average F1 score outperformed the baseline classifiers (Majority, random). Further, one can notice that, on average, the classifiers had an easier time classifying higher classes (4,5) rather than lower classes (especially 2,3), probably due to the finer line between class 2 and 3 vs. class 4 and 5.

On test data, Linear SVM outperformed Naïve Bayes and Decision Trees for frequency bag-of-words sentiment classification. Further, comparing binary and frequency bag-of-words representation, binary outperformed frequency representation by between 1% to 13% (when comparing the same classifier’s test performance for each representation). This could be due to the fact that the increased frequency of less significant words (like ‘the’) is taken into account for the latter representation.

# Sentiment Classification – IMDB

For the IMDB sentiment classification, as a baseline, the performance of a uniformly random classifier is outlined below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classifier (Set) | Class 0 | Class 1 | | Ave |
| Random (Test) | 0.5006 | 0.4973 | | 0.4989 |
| Random (Train) | 0.5049 | 0.4995 | | 0.5022 |
| Random (Valid) | 0.5001 | | 0.4937 | 0.4969 |

### Binary Bag-of-Words - IMDB

Bernoulli Naïve Bayes

For Bernoulli Naïve Bayes, the hyper-parameter being tuned was α, the additive (Laplace) smoothing parameter. The details of the hyper-parameter tuning are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Hyper-parameter | Range | Best Value | Notes |
| α | (0, 10] |  | α was incremented by hundredths (0.01,0.02…) |

In the table below, the F1 performance measures (test, training, validation) for Bernoulli Naïve Bayes with the best hyper-parameter values are provided:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Set | Class 0 | Class 1 | | Ave |
| Test Set | 0.8368 | 0.8376 | | 0.8372 |
| Training Set | 0.8583 | 0.8617 | | 0.8600 |
| Validation Set | 0.8591 | | 0.8622 | 0.8606 |

Linear SVM

For Linear SVM, the hyper-parameter being tuned was the penalty parameter C of the error term. The details of the hyper-parameter tuning are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Hyper-parameter | Range | Best Value | Notes |
| Loss function, L | *{squared\_hinge}* |  | L1 and square\_hinge, hinge and dual=False are incompatible; |
| Penalty parameter, C | (0,4] | *C = 0.01* | C was incremented by 0.01, so C = 0.01, 0.02, 0.03… |
| Penalization, P | *P = L2* | *P = L2* | P = L1 wasn’t considered due to incompatibility with squared hinge |
| Dual | *Dual = False* | *Dual = False* | Less features than data points, so solve primal |

In the table below, the F1 performance measures (test, training, validation) for Linear SVM with the best hyper-parameter values are provided:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Set | Class 0 | Class 1 | | Ave |
| Test Set | 0.8718 | 0.8735 | | 0.8727 |
| Training Set | 0.9312 | 0.9322 | | 0.9317 |
| Validation Set | 0.9325 | | 0.9332 | 0.9328 |

Decision Trees

For Decision Tree, the hyper-parameters being tuned were: *criterion, splitter, max depth,* and *min\_sample\_leafs*. The details of the hyper-parameter tunings are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Hyper-parameter | Range | Best Value | Notes |
| Criterion, C | {gini, entropy} |  |  |
| Splitter, S | {best, random} | *S = random* |  |
| Max Depth, D | *D*  *{None,10,20,30,40*  *,50,60,70,80,90}* | *D = 20* |  |
| Min Sample Leaf, M | *M*  *{1,2,3,4,5,6,7,8,9,*  *10,20,30,40,50}* | *M = 40* |  |

In the table below, the F1 performance measures (test, training, validation) for Decision Tree with the best hyper-parameter values are provided:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Set | Class 0 | Class 1 | | Ave |
| Test Set | 0.7219 | 0.7179 | | 0.7199 |
| Training Set | 0.7305 | 0.7216 | | 0.7261 |
| Validation Set | 0.7285 | | 0.7211 | 0.7248 |

### Comments

Each classifier’s average F1 score outperformed the baseline classifier (random).

On test data, Linear SVM slimly outperformed Bernoulli Naïve Bayes, and both classifiers widely outperformed the Decision Tree classifier. On the whole, the classifiers performed more favorably in the IMDB binary representation classification than the Yelp classification.

### Frequency Bag of Words - IMDB

Gaussian Naïve Bayes

For Gaussian Naïve Bayes, there were no hyper-parameters to tune.

In the table below, the F1 performance measures (test, training, validation) for Gaussian Naïve Bayes are provided:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Set | Class 0 | Class 1 | | Ave |
| Test Set | 0.6518 | 0.6686 | | 0.6602 |
| Training Set | 0.7930 | 0.8104 | | 0.8017 |
| Validation Set | 0.7923 | | 0.8122 | 0.8022 |

Linear SVM

For Linear SVM, the hyper-parameter being tuned was the penalty parameter C of the error term. The details of the hyper-parameter tuning are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Hyper-parameter | Range | Best Value | Notes |
| Loss function, L | *{squared\_hinge}* |  | L1 and square\_hinge, hinge and dual=False are incompatible; |
| Penalty parameter, C | (0,10] | *C = 10.0* | C was incremented by 0.5, so C = 0.5, 1.0, 1.5… |
| Penalization, P | *P = L2* | *P = L2* | P = L1 wasn’t considered due to incompatibility with squared hinge |
| Dual | *Dual = False* | *Dual = False* | Less features than data points, so solve primal |

In the table below, the F1 performance measures (test, training, validation) for Linear SVM with the best hyper-parameter values are provided:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Set | Class 0 | Class 1 | | Ave |
| Test Set | 0.8671 | 0.8684 | | 0.8677 |
| Training Set | 0.8836 | 0.8852 | | 0.8844 |
| Validation Set | 0.8804 | | 0.8824 | 0.8814 |

Decision Trees

For Decision Tree, the hyper-parameters being tuned were: *criterion, splitter, max depth,* and *min\_sample\_leafs*. The details of the hyper-parameter tunings are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Hyper-parameter | Range | Best Value | Notes |
| Criterion, C | {gini, entropy} |  |  |
| Splitter, S | {best, random} | *S = random* |  |
| Max Depth, D | *D*  *{None,10,20,30,40*  *,50,60,70,80,90}* | *D = 20* |  |
| Min Sample Leaf, M | *M*  *{1,2,3,4,5,6,7,8,9,*  *10,20,30,40,50}* | *M = 1* |  |

In the table below, the F1 performance measures (test, training, validation) for Decision Tree with the best hyper-parameter values are provided:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Set | Class 0 | Class 1 | | Ave |
| Test Set | 0.6904 | 0.7474 | | 0.7189 |
| Training Set | 0.7882 | 0.8309 | | 0.8096 |
| Validation Set | 0.7861 | | 0.8279 | 0.8070 |

### Comments

Each classifier’s average F1 score outperformed the baseline classifier (random).

On test data, Linear SVM outperformed Naïve Bayes and Decision Trees for frequency bag-of-words sentiment classification. Further, comparing binary and frequency bag-of-words representation, the Linear SVM classifier and the Decision Tree classifier both faired similarly between the representations, whereas the Gaussian Naïve Bayes classifier did around 15% worse than the Bernoulli Naïve Bayes.

Thus, the classifiers on the whole performed better with the binary representation than the frequency bag-of-words representation. This could be due to words of lesser significance, but higher frequency, having more influence in the classification.

Between the datasets, the relative performance of each of the classifiers stayed constant. The Linear SVM consistently outperformed the Naïve Bayes and Decision Tree classifier for both representations (binary and frequency) of the input data (using the F1 score on the Test data set as the metric of comparison).