Diego Valdes

IST 707

Home Work 5

5/1/2019

**Introduction**

For as long as humans have been on the Earth, they have tried to use tools to improve the lifestyle of the species. It started with bone weapons for hunting, domesticating animals for labor and hunting, and then advances moved to more advanced materials and concepts. Bone weapons evolved to stone and then metal; economy changed from trading farm goods and livestock to using valuable metals, then coins, and promissory notes. The tools became more elaborate with time, allowing mankind to create cities, travel the world, and explore the heavens.

While these tools are impressive to most, in the 21st century, humans have moved to designing a different type of tool—a smart tool. Silicon is the new metal and the computer is the new dog. The computer is the tool that has opened immense possibilities and new avenues of creativity for the species. It is quite possibly the most adaptable tool ever designed. It has gotten more powerful as it has gotten smaller, it has evolved to do differing tasks, it is responsible for whole industries rising, it has changed the lives and interactions of humans, and it has made the impossible easy to accomplish. It can even mimic its creator.

Part of the machine learning to mimic humans is recognizing images. Image processing is useful for more advanced causes, like identifying cancerous cells, but it can also be used to find Waldo—clearly the most valuable use of the processing power of a computer. This ability while can be used for the most worthy endeavors, or the simplest, has practical everyday applications as well. Being able to identify human handwriting, with all of the intricacies that are present between each writer, can be as valuable to a corporation as the documents that could be processed at record speeds if done accurately.

**Analysis**

**About the Data**

The dataset originates from the Kaggle Digit Recognizer competition, with the goal of recognizing handwriting samples. The original datasets have 42,000 (train) and 28,000 (test) observations with 785 attributes. Each attribute represents a pixel on a 28 x 28 grid, where a number greater than 0 represents that pixel is part of the handwriting sample. Each pixel attribute is labeled ‘pixelNNN’ where NNN is the assigned number starting from 0 - 783 (fig 1.1).

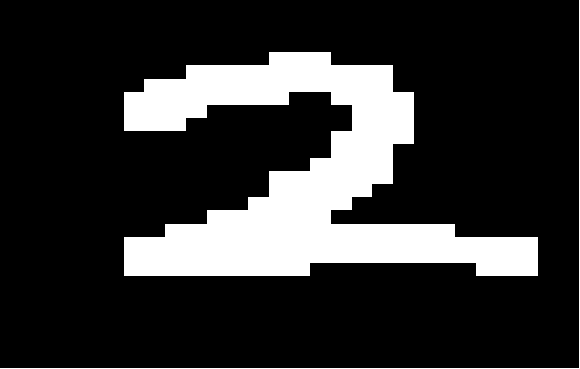
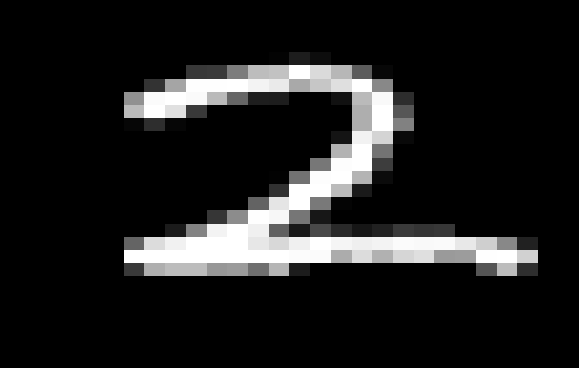
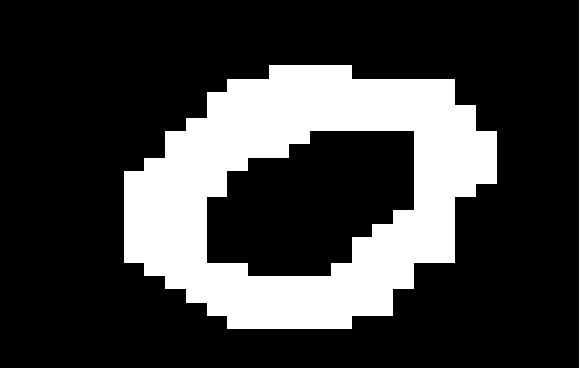
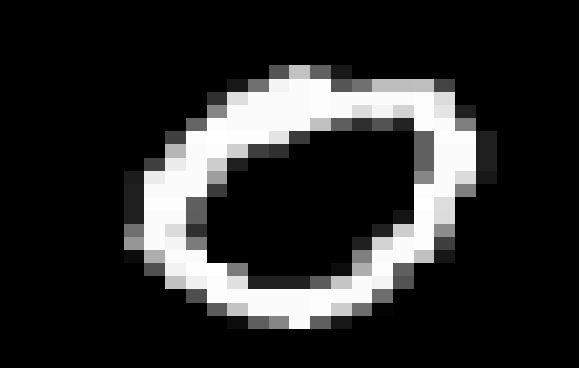
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 |
| 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 |
| 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 |
| 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 | 195 |
| 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 | 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 |
| 252 | 253 | 254 | 255 | 256 | 257 | 258 | 259 | 260 | 261 | 262 | 263 | 264 | 265 | 266 | 267 | 268 | 269 | 270 | 271 | 272 | 273 | 274 | 275 | 276 | 277 | 278 | 279 |
| 280 | 281 | 282 | 283 | 284 | 285 | 286 | 287 | 288 | 289 | 290 | 291 | 292 | 293 | 294 | 295 | 296 | 297 | 298 | 299 | 300 | 301 | 302 | 303 | 304 | 305 | 306 | 307 |
| 308 | 309 | 310 | 311 | 312 | 313 | 314 | 315 | 316 | 317 | 318 | 319 | 320 | 321 | 322 | 323 | 324 | 325 | 326 | 327 | 328 | 329 | 330 | 331 | 332 | 333 | 334 | 335 |
| 336 | 337 | 338 | 339 | 340 | 341 | 342 | 343 | 344 | 345 | 346 | 347 | 348 | 349 | 350 | 351 | 352 | 353 | 354 | 355 | 356 | 357 | 358 | 359 | 360 | 361 | 362 | 363 |
| 364 | 365 | 366 | 367 | 368 | 369 | 370 | 371 | 372 | 373 | 374 | 375 | 376 | 377 | 378 | 379 | 380 | 381 | 382 | 383 | 384 | 385 | 386 | 387 | 388 | 389 | 390 | 391 |
| 392 | 393 | 394 | 395 | 396 | 397 | 398 | 399 | 400 | 401 | 402 | 403 | 404 | 405 | 406 | 407 | 408 | 409 | 410 | 411 | 412 | 413 | 414 | 415 | 416 | 417 | 418 | 419 |
| 420 | 421 | 422 | 423 | 424 | 425 | 426 | 427 | 428 | 429 | 430 | 431 | 432 | 433 | 434 | 435 | 436 | 437 | 438 | 439 | 440 | 441 | 442 | 443 | 444 | 445 | 446 | 447 |
| 448 | 449 | 450 | 451 | 452 | 453 | 454 | 455 | 456 | 457 | 458 | 459 | 460 | 461 | 462 | 463 | 464 | 465 | 466 | 467 | 468 | 469 | 470 | 471 | 472 | 473 | 474 | 475 |
| 476 | 477 | 478 | 479 | 480 | 481 | 482 | 483 | 484 | 485 | 486 | 487 | 488 | 489 | 490 | 491 | 492 | 493 | 494 | 495 | 496 | 497 | 498 | 499 | 500 | 501 | 502 | 503 |
| 504 | 505 | 506 | 507 | 508 | 509 | 510 | 511 | 512 | 513 | 514 | 515 | 516 | 517 | 518 | 519 | 520 | 521 | 522 | 523 | 524 | 525 | 526 | 527 | 528 | 529 | 530 | 531 |
| 532 | 533 | 534 | 535 | 536 | 537 | 538 | 539 | 540 | 541 | 542 | 543 | 544 | 545 | 546 | 547 | 548 | 549 | 550 | 551 | 552 | 553 | 554 | 555 | 556 | 557 | 558 | 559 |
| 560 | 561 | 562 | 563 | 564 | 565 | 566 | 567 | 568 | 569 | 570 | 571 | 572 | 573 | 574 | 575 | 576 | 577 | 578 | 579 | 580 | 581 | 582 | 583 | 584 | 585 | 586 | 587 |
| 588 | 589 | 590 | 591 | 592 | 593 | 594 | 595 | 596 | 597 | 598 | 599 | 600 | 601 | 602 | 603 | 604 | 605 | 606 | 607 | 608 | 609 | 610 | 611 | 612 | 613 | 614 | 615 |
| 616 | 617 | 618 | 619 | 620 | 621 | 622 | 623 | 624 | 625 | 626 | 627 | 628 | 629 | 630 | 631 | 632 | 633 | 634 | 635 | 636 | 637 | 638 | 639 | 640 | 641 | 642 | 643 |
| 644 | 645 | 646 | 647 | 648 | 649 | 650 | 651 | 652 | 653 | 654 | 655 | 656 | 657 | 658 | 659 | 660 | 661 | 662 | 663 | 664 | 665 | 666 | 667 | 668 | 669 | 670 | 671 |
| 672 | 673 | 674 | 675 | 676 | 677 | 678 | 679 | 680 | 681 | 682 | 683 | 684 | 685 | 686 | 687 | 688 | 689 | 690 | 691 | 692 | 693 | 694 | 695 | 696 | 697 | 698 | 699 |
| 700 | 701 | 702 | 703 | 704 | 705 | 706 | 707 | 708 | 709 | 710 | 711 | 712 | 713 | 714 | 715 | 716 | 717 | 718 | 719 | 720 | 721 | 722 | 723 | 724 | 725 | 726 | 727 |
| 728 | 729 | 730 | 731 | 732 | 733 | 734 | 735 | 736 | 737 | 738 | 739 | 740 | 741 | 742 | 743 | 744 | 745 | 746 | 747 | 748 | 749 | 750 | 751 | 752 | 753 | 754 | 755 |
| 756 | 757 | 758 | 759 | 760 | 761 | 762 | 763 | 764 | 765 | 766 | 767 | 768 | 769 | 770 | 771 | 772 | 773 | 774 | 775 | 776 | 777 | 778 | 779 | 780 | 781 | 782 | 783 |

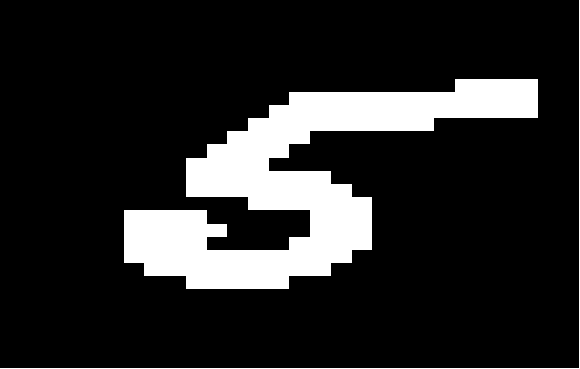
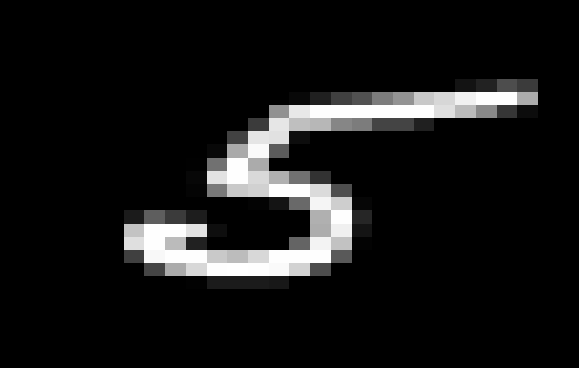
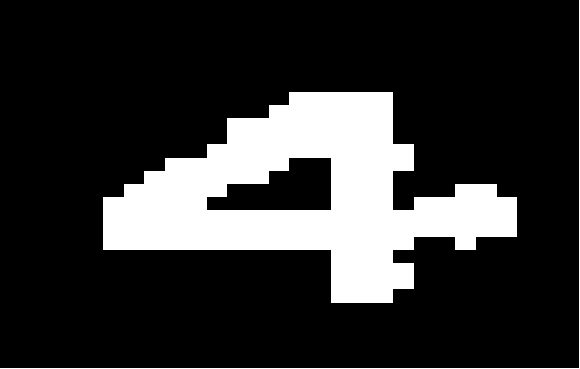
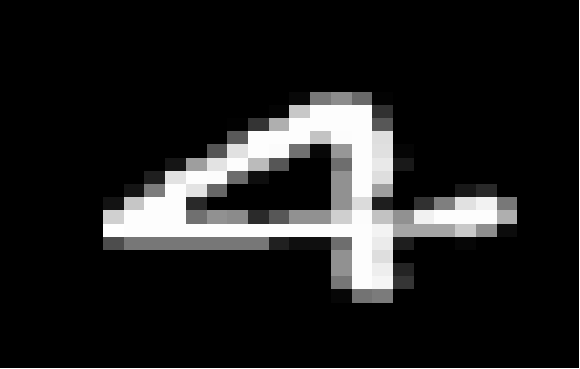
Fig 1.1: 28x28 matrix mapped with column numbers

The datasets used for the analysis consisted of 10% of the original data, sampled from each 10th element, ie: 10, 20, 30, … These sets had 1400 (train) and 1000 (test) observations (digit\_orig). The dataset had no missing values and required no cleaning. The first column, label, identifies the digit. This column was transformed into a factor. All other columns were left as numbers.

The range for pixels is 0 – 255. An alternative dataset (digit\_binary) was created where values above 0 where converted to 1. The original number range represents the color variation in the sample, but this was done to analyze whether a binary analysis would prove to provide a better result, as the goal of the analysis is to properly identify digits, not the various colors possible in creating the digit.

A sample of each digit (0 – 9) was rearranged into a 28x28 grid to plot the sample. This was done for both digit\_orig and digit\_binary datasets (fig 1.2). Both samples can be clearly identified as there respective digit, but the digit\_binary samples produce more defined samples.





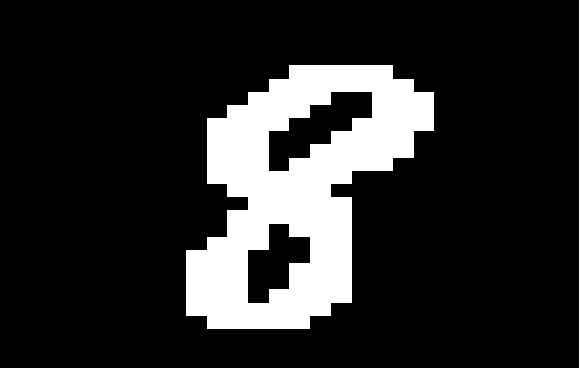
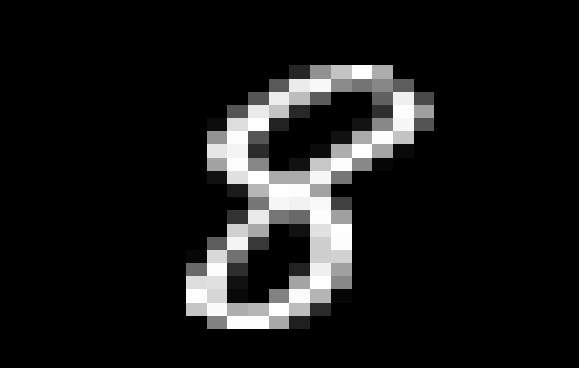
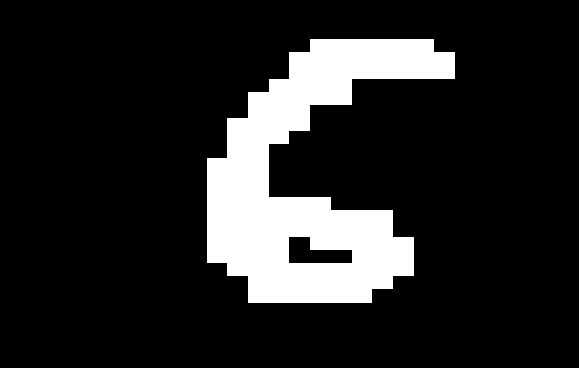
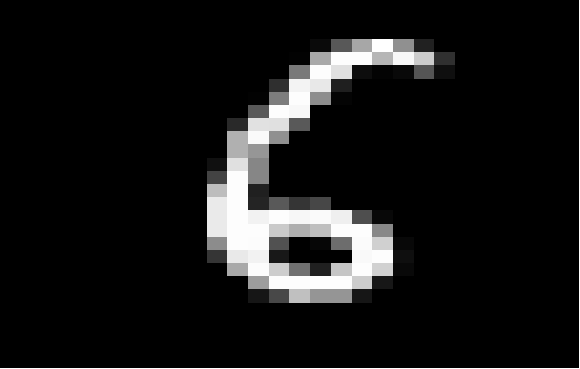


Fig 1.2: plots of digits (0,2,4,5,6,8) generated by transforming sample row to 28x28 matrix

The datasets were divided into a training sets with 70% of randomly sampled data, and test sets with the remaining 30%. There is also a test set where the label column is filled with ‘?’ as part of the Kaggle competition. This set is good for a submission, but can’t be used for testing the accuracy of the model.

**Models**

**Decision Tree (rpart::rpart)**

The decision tree model for both digit\_orig and digit\_binary went through multiple modifications before reaching an optimal solution. The formula for the model was label as a function of all of the pixels (label ~.). Every attribute was included as each pixel value is part of the whole image.

Pruning the tree included determining the min node split, complexity parameter (cp), and max depth. The cp value was immediately determined to be 0. Any value above this created poor performing models. The min node split varied between 2 and 9. Overall model performance on the test data increased with each increase in the value. No performance increase was observed with a value above 9. The max depth was set to minimize overfitting. The larger the tree, when the model was run on the training set, the accuracy was 100%. This produced accuracy results below 50% on an independent testing set. Depths lesser than 10 didn’t improve those results. These observations were consistent along both datasets. The final values:

* min split, 9
* cp, 0
* max depth, 10

**Naïve Bayes (e1071::naiveBayes)**

The Naïve Bayes model didn’t go through many modifications for digit\_orig or digit\_binary datasets. None of the other attributes for the algorithm didn’t apply to these datasets. One of the values would allow percentages to be substituted instead of values; while this would be a useful change in some analysis, for identifying handwriting examples, it wouldn’t be the correct approach. The formula for the model was label as a function of all of the pixels (label ~.). Every attribute was included as each pixel value is part of the whole image.

The only attribute that was used was for Laplace smoothing. Providing the default value of 0 produced less accurate results when running the test data. A value of 1 was provided, as it produced the best results against the test set.

**Results**

**Decision Tree (rpart::rpart) – dataset: digit\_orig**

Model Statistics on training set:

* Acc: 88%
* 95% CI: (0.8674, 0.9075)
* NIR: 0.1265
* Kappa: 0.8763

Model Statistics on test set:

* Acc: 68%
* 95% CI: (0.6365, 0.7276)
* NIR: 0.1262
* Kappa: 0.6478

**Decision Tree (rpart::rpart) – dataset: digit\_binary**

Model Statistics on training set:

* Acc: 86%
* 95% CI: (0.8423, 0.8861)
* NIR: 0.1265
* Kappa: 0.8501

Model Statistics on test set:

* Acc: 65%
* 95% CI: (0.6096, 0.7025)
* NIR: 0.1262
* Kappa: 0.6189

For both decision tree models, the model was ran against the training set and produced high results for predicting the digits. This remained consistent with random samples of training data across multiple model creations. Once test data was used, all statistics dropped, only NIR (No Information Rate) remained constant. The digit\_orig model consistently performed better than digit\_binary, 68% vs 65%, (fig 2.1a, 2.1b).

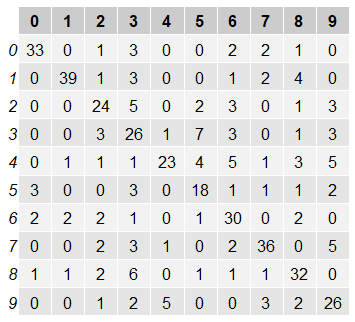
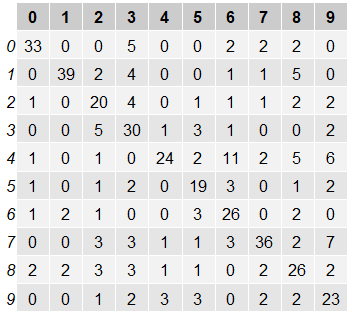
 

Fig 2.1a: confusion matrix – training data Fig 2.1b: confusion matrix – testing data

Although digit\_orig decision tree performed better, a two sample t-test was performed to determine if the percentage difference was statistically significant. The null hypothesis: there is no difference in the results based on transforming the values in the dataset to be binary. Based on the results of the test, a p-value of 0.3573 is not significant at the 0.05 level and the null hypothesis cannot be rejected. The difference in percentages between the two models isn’t significant enough to determine that one dataset is better suited to create a model to determine the correct digit.

Two Sample t-test

* T-statistic: 0.921
* DF: 838
* Two-tailed probability: .3573

**Naïve Bayes (e1071::naiveBayes) – dataset: digit\_orig**

Model Statistics on training set:

* Acc: 57%
* 95% CI: (0.5439, 0.6067)
* NIR: 0.1265
* Kappa: 0.5261

Model Statistics on test set:

* Acc: 53%
* 95% CI: (0.4891, 0.5865)
* NIR: 0.1262
* Kappa: 0.4836

**Naïve Bayes (e1071::naiveBayes) – dataset: digit\_binary**

Model Statistics on training set:

* Acc: 65%
* 95% CI: (0.6202, 0.6809)
* NIR: 0.1265
* Kappa: 0.6107

Model Statistics on test set:

* Acc: 63%
* 95% CI: (0.5901, 0.6841)
* NIR: 0.1262
* Kappa: 0.5956

In the Naïve Bayes functions, the difference between the model runs with training data and test data did not produce significant differences as were observed in the decision tree models. Like the decision tree statistics, the only value that was consistent across training and test runs was NIR. Unlike the decision tree, model runs using digit\_binary model produced the higher accuracy 63% vs 53%, (fig 2.2a, 2.2b).

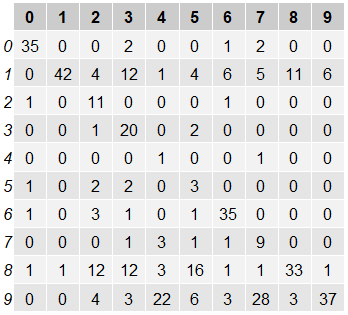
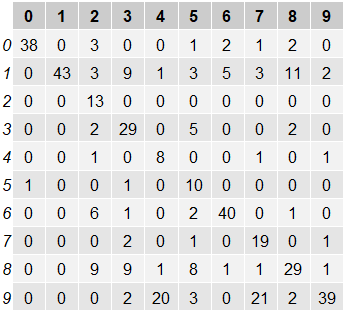
 

Fig 2.2a: confusion matrix – training data Fig 2.2b: confusion matrix – testing data

A two sample t-test was performed to determine if the accuracy percentage difference between models was statically significant. The null hypothesis: there is no difference in the results based on transforming the values in the dataset to be binary. Based on the results of the test, a p-value of 0.0034 is significant at the 0.05 level and the null hypothesis can be rejected. The difference between the two models is significant enough to determine that the digit\_binary dataset is better suited to produce a Naïve Bayes model to make a determination on the handwriting samples.

Two Sample t-test

* T-statistic: 2.936
* DF: 838
* Two-tailed probability: 0.0034

**Decision Tree vs. Naïve Bayes**

There was no statistical significance between the decision tree models using digit\_orig and digit\_binary datasets, and there was a statistical significance in the Naïve Bayes models that were created using these datasets. To examine if either Decision Tree model was statistically better than the Naïve Bayes model using the digit\_binary dataset.

Two Sample t-test – decision tree (digit\_orig) vs naïve bayes (digit\_binary)

* T-statistic: 1.524
* DF: 838
* Two-tailed probability: 0.1278

Two Sample t-tes t– decision tree (digit\_binary) vs naïve bayes (digit\_binary)

* T-statistic: 0.604
* DF: 838
* Two-tailed probability: 0.5461

The null hypothesis for both tests was that there is no difference in the results based on the model being used. In the case of digit\_orig and digit\_binary model comparisons, this also extended to the transformation applied to the data. Both tests didn’t provide a significant p-value at 0.05 significance to reject the null hypothesis. While an examination of the accuracy of all the models would leave an observer to believe the decision tree (digit\_orig) performed better due to its higher accuracy, it did not perform good enough to make it the clear choice statistically.

**Conclusion**

The technology to be able to accurately read handwriting samples can be valuable to a multitude of organizations; especially those that have a collection of handwritten documents. Before, a company would need an army of employees just to read through and input the data into a system. Replacing this workforce with computers would be cost effective and increase the time to process documents.

Unfortunately, naïve bayes and decision trees as they were used in this analysis proves that on their own they are not good enough to provide an accurate identification process. The best accuracy of any model on its own is approximately 60%. A better way to look at this is that the model is incorrect 40% of the time. There is too much variation in identifying handwriting between all the samples to provide better accuracy.

More samples are not likely to improve these results. There isn’t enough pixel consistency within each digit classification to improve on the predictions. As an example, in a 28x28 grid, the starting point of the digit ‘1’ won’t always have the same starting pixel, range of pixels, and traverse the same pixels. That digit can take multiple shapes and sizes (1, 1, 1, 1, 1) and can have similar strokes to other digits (7, 7, 7, 7, 7).

The model needs supplemental support. There are a few forms that this can be accomplished; humans checking on the models result, using multiple models in a staggered fashion to increase accuracy, or using different machine learning algorithms that are better suited to determine the correct digits. Another process that could be implemented with letters are for the models implement the rules of language. As an example, if the first letter of a word is ‘R’, what is the percentage that the next letter is a vowel vs a consonant? If the percentage for a consonant is higher, than what are the percentages that the letter is a, e, i, o, u? This additional evaluation method, in conjunction with the model, could greatly improve accuracy, but wouldn’t be effective in simple digit prediction.