Jefferson Parker

CS 677

April 3, 2022

Homework 3, Written Answers

**Question 1.**

* 1. - See code in japarker\_hw3\_1.py.
  2. - The table of mean and standard deviation for each feature, F1 – F4 is:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Class | F1 | | F2 | | F3 | | F4 | |
| mean | std | mean | std | mean | std | mean | std |
| Class 0 | 2.28 | 2.02 | 4.26 | 5.14 | 0.8 | 3.24 | -1.15 | 2.13 |
| Class 1 | -1.87 | 1.88 | -0.99 | 5.4 | 2.15 | 5.26 | -1.25 | 2.07 |
| All | 0.43 | 2.84 | 1.92 | 5.87 | 1.4 | 4.31 | -1.19 | 2.1 |

* 1. – There are some patterns obvious in the class data:
* Class 0 (real notes) on average have positive values for F1 and F2, but negative values for F3.
* Class 1 (fake notes) on average have negative values for F1 and F2, but positive values for F3.
* F4 does not seem to differentiate between either class.

**Question 2.**

2.1 – See code in japarker\_hw3\_2.py and PDF files japarker\_hw3\_fake\_bills.pdf and japarker\_hw3\_good\_bills.pdf.

2.2 – From visual inspection of the plots I selected a simple classifier of: F1 >= 0 and F2 >= 5 and F3 <= 5 to assign ‘good’ bills and otherwise assign ‘false’ bills. This was implemented in JparkerHw3Helper.py as function simple\_simple\_classifier.

2.3 – See code in japarker\_hw3\_2.py for implementation of the simple classifier.

2.4 – See code in japarker\_hw3\_2.py for calculation of classifier metrics.

2.5 – The performance metrics for the simple classifier are:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TP | FP | TN | FN | Acc | TPR | TNR |
| 168 | 1 | 304 | 213 | 0.688 | 0.441 | 0.997 |

2.6 – The simple classifier performs better than random with an accuracy of ~69%. While it is probably not enough for actual use in identifying fake bills it is better than randomly splitting them.

**Question 3.**

3.1 – See code in japarker\_hw3\_3.py for implementation of k-NN classifier and calculation of accuracy (k\_accuracy). Interestingly (oddly?) the accuracy was 1 for all k >= 5.

3.2 – The plot of k vs. accuracy is:

Chart, line chart

Description automatically generated

3.3 – For the optimal k\* = 5, the performance metrics are:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TP | FP | TN | FN | accuracy | TPR | TNR |
| 381 | 0 | 305 | 0 | 1 | 1 | 1 |

3.4 The k-NN classifier is better than the simple classifier in all measures.

3.5 – For a bill with feature values equal to the last four digits of my BUID (8928) the predicted class label from both the simple classifier and the k-NN (5) classifier is “good”.

**Question 4.**

4.1 – For k\* = 5, dropping each of the features f1 – f4, the performance metrics are:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Feature | TP | FP | TN | FN | accuracy | TPR | TNR |
| F1 | 367 | 13 | 292 | 14 | 0.961 | 0.963 | 0.957 |
| F2 | 372 | 5 | 300 | 9 | 0.980 | 0.976 | 0.984 |
| F3 | 370 | 7 | 298 | 11 | 0.974 | 0.971 | 0.977 |
| F4 | 380 | 0 | 305 | 1 | 0.999 | 0.997 | 1.000 |

4.2 – Dropping any of the four features did not increase the accuracy over using all four. However, the accuracy for k\* = 5 using all four features had an accuracy of 1 (100%).

4.3 – Feature F1 contributed the greatest loss of accuracy, from 1 to 0.961. This would suggest that it is the most important feature for classification.

4.4 – Feature F4 contributed the least loss of accuracy, from 1 for all features to 0.998, suggesting that it is the least important feature for classification.

**Question 5.**

5.1 – See code in japarker\_hw3\_5.py for implementation of logistic regression.

5.2 – The performance metrics for logistic regression are:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TP | FP | TN | FN | accuracy | TPR | TNR |
| 374 | 2 | 303 | 7 | 0.987 | 0.982 | 0.993 |

5.3 – Logistic regression (accuracy = 0.987) performed better than the simple classifier of question 2.5 (accuracy 0.688). My understanding is that the logistic regression classifier would make a correct assignment 30% more frequently than the simple classifier.

5.4 – The logistic regression classifier DID NOT perform better than the k-NN classifier with k\* = 5 (accuracy = 1.0). They were both very good, but strictly comparing the numbers the logistic regression classifier did just slightly worse.

5.5 – For a bill with features F1 – F4 corresponding to my BUID (8928) the class label predicted is 0 (good bill). This is the same label predicted by k-NN classification.

**Question 6.**

6.1 – Applying logistic regression to the data set, serially dropping each of the four features F1 – F4 had performance of:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Feature | TP | FP | TN | FN | accuracy | TPR | TNR |
| F1 | 310 | 52 | 253 | 71 | 0.821 | 0.814 | 0.830 |
| F2 | 353 | 41 | 264 | 28 | 0.899 | 0.927 | 0.866 |
| F3 | 349 | 52 | 253 | 32 | 0.878 | 0.916 | 0.830 |
| F4 | 374 | 2 | 303 | 7 | 0.987 | 0.982 | 0.993 |

6.2 – The accuracy with logistic regression did improve by 0.05 when feature F4 was removed. Previously feature F4 was found to be not differential (mean values in Q1, visual inspection in Q2) and having the smallest impact on accuracy when dropped from kNN (4.1). For logistic regression, inclusion of F4 WORSENED performance.

6.3 – Again, removing feature F1 contributed the greatest loss to prediction accuracy (0.982 to 0.821 without F1).

6.4 – As mentioned, removing feature F4 contributed the least to loss of accuracy to such a degree that accuracy improved.

6.5 – The relative significance of features (order of improvement when dropped) was the same for both logistic regression and k-NN classification: F1 > F3 > F2 > F4.