Jupyter Notebook: <https://github.com/ben12385/DS4400-HW2>

Readme: In the Github

**Problem 1 [Logistic regression]**

(a) Split the original data into 75% for training and 25% for testing. Choose the training set at random and ensure that the ratio of SPAM examples in the training set is close to the ratio of 39.4% SPAM examples in the entire dataset. Train a logistic regression model on the training set and output the following on the testing set:

Confusion matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Predicted Class | | | |
| Actual Class |  | Spam | Not Spam |
| Spam | 40 | 43 |
| Not Spam | 28 | 668 |

True Positives - 410

False Positives - 28

True Negatives - 668

False Negatives - 43

Accuracy – 94%

Error – 6%

Precision – 0.94

Recall – 0.91

F1 score – 0.92

(b) Print the coefficients of the features in the model. Which features contribute mostly to the prediction? Which ones are positively correlated, and which ones are negatively correlated with the SPAM class?

|  |  |
| --- | --- |
| Coefficents | Positive or Negative Correlated |
| word\_freq\_make | Negative |
| word\_freq\_address | Negative |
| word\_freq\_all | Positive |
| word\_freq\_3d | Positive |
| word\_freq\_our | Positive |
| word\_freq\_over | Positive |
| word\_freq\_remove | Positive |
| word\_freq\_internet | Positive |
| word\_freq\_order | Positive |
| word\_freq\_mail | Positive |
| word\_freq\_receive | Positive |
| word\_freq\_will | Negative |
| word\_freq\_people | Negative |
| word\_freq\_report | Positive |
| word\_freq\_addresses | Positive |
| word\_freq\_free | Positive |
| word\_freq\_business | Positive |
| word\_freq\_email | Positive |
| word\_freq\_you | Positive |
| word\_freq\_credit | Positive |
| word\_freq\_your | Positive |
| word\_freq\_font | Positive |
| word\_freq\_000 | Positive |
| word\_freq\_money | Positive |
| word\_freq\_hp | Negative |
| word\_freq\_hpl | Negative |
| word\_freq\_george | Negative |
| word\_freq\_650 | Positive |
| word\_freq\_lab | Negative |
| word\_freq\_labs | Negative |
| word\_freq\_telnet | Negative |
| word\_freq\_857 | Negative |
| word\_freq\_data | Negative |
| word\_freq\_415 | Positive |
| word\_freq\_85 | Negative |
| word\_freq\_technology | Positive |
| word\_freq\_1999 | Negative |
| word\_freq\_parts | Negative |
| word\_freq\_pm | Negative |
| word\_freq\_direct | Negative |
| word\_freq\_cs | Negative |
| word\_freq\_meeting | Negative |
| word\_freq\_original | Negative |
| word\_freq\_project | Negative |
| word\_freq\_re | Negative |
| word\_freq\_edu | Negative |
| word\_freq\_table | Negative |
| word\_freq\_conference | Negative |
| char\_freq\_; | Negative |
| char\_freq\_( | Negative |
| char\_freq\_[ | Negative |
| char\_freq\_! | Positive |
| char\_freq\_$ | Positive |
| char\_freq\_# | Positive |
| capital\_run\_length\_average | Negative |
| capital\_run\_length\_longest | Positive |
| capital\_run\_length\_total | Positive |

The features that contribute the most to the prediction are “char freq $” and “word freq remove”.

(c) Vary the decision threshold T ∈ {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} and report for each value the model accuracy. Plot a graph of the accuracy of the model as a function of the threshold.

|  |  |
| --- | --- |
| Threshold | Accuracy |
| 0.1 | 0.82245 |
| 0.2 | 0.89469 |
| 0.3 | 0.92515 |
| 0.4 | 0.93995 |
| 0.5 | 0.93821 |
| 0.6 | 0.9295 |
| 0.7 | 0.91384 |
| 0.8 | 0.88947 |
| 0.9 | 0.8503 |

**Problem 2 [Comparing classifiers]**

(a) Experiment with different values of k for kNN and report 2 metrics: accuracy and error. Choose the value of k that gives the highest accuracy.

|  |  |
| --- | --- |
| kNN Value | Accuracy |
| 2 | 0.80418 |
| 3 | 0.80505 |
| 4 | 0.79547 |
| 5 | 0.80070 |
| 6 | 0.79112 |
| 7 | 0.79547 |
| 8 | 0.77459 |
| 9 | 0.77459 |

The value of 3 is chosen for k because it has the highest accuracy.

(b) Print the accuracy and error metrics for all 4 classifiers. Which model is performing best? Which one is performing worst? Write down some observations.

|  |  |  |
| --- | --- | --- |
| Classifier | Accuracy | Error |
| Logistic Regression | 0.93 | 0.07 |
| LDA | 0.89 | 0.11 |
| kNN | 0.81 | 0.19 |
| Decision Tree | 0.91 | 0.09 |

The model that performs the best is logistic regression while the worst is kNN. LDA did not do as well is probably due to the fact that the features do not have a Gaussian distribution. The decision tree performing with the second highest accuracy is surprising because decision tree tends to overfit the training data and no known pruning was done on the tree.

(c) Generate a graph that includes 4 ROC curves (one for each of the 4 classifiers). Compute the Area Under the Curve (AUC) metric for all 4 classifiers.

|  |  |  |
| --- | --- | --- |
| FPR | TPR | Threshold |
| 0 | 0 | 0 |
| 0.2945 | 0.9890 | 0.1 |
| 0.1566 | 0.9647 | 0.2 |
| 0.1020 | 0.9360 | 0.3 |
| 0.0675 | 0.9272 | 0.4 |
| 0.0460 | 0.8896 | 0.5 |
| 0.0374 | 0.8477 | 0.6 |
| 0.0316 | 0.7947 | 0.7 |
| 0.0244 | 0.7196 | 0.8 |
| 0.0144 | 0.6225 | 0.9 |
| 1 | 1 | 1 |

AUC = 0.9727

|  |  |  |
| --- | --- | --- |
| FPR | TPR | Threshold |
| 0 | 0 | 0 |
| 0.2514 | 0.9470 | 0.1 |
| 0.1178 | 0.9205 | 0.2 |
| 0.0632 | 0.9051 | 0.3 |
| 0.0460 | 0.8565 | 0.4 |
| 0.0388 | 0.8035 | 0.5 |
| 0.0359 | 0.7351 | 0.6 |
| 0.0259 | 0.6689 | 0.7 |
| 0.0158 | 0.5960 | 0.8 |
| 0.0101 | 0.4834 | 0.9 |
| 1 | 1 | 1 |

AUC = 0.9570

|  |  |  |
| --- | --- | --- |
| FPR | TPR | Threshold(# of neighbours) |
| 0 | 0 | 0 |
| 0.1451 | 0.7241 | 3 |
| 1 | 1 | 1 |

AUC = 0.7731

|  |  |  |
| --- | --- | --- |
| FPR | TPR | Threshold |
| 0 | 0 | 0 |
| 0.072 | 0.881 | 1 |
| 1 | 1 | 1 |

AUC = 0.9045

**Problem 3 [kNN]**

(c) Pick several values of k (the same ones you picked in Problem 2) and print the accuracy and error metrics on the test set using your implementation of the kNN classifier.

|  |  |
| --- | --- |
| kNN Value | Accuracy |
| 2 | 0.80244 |
| 3 | 0.80940 |
| 4 | 0.78590 |
| 5 | 0.78938 |
| 6 | 0.77981 |
| 7 | 0.78590 |
| 8 | 0.77633 |
| 9 | 0.78677 |

(d) Compare the results obtained by your implementation with those obtained in Problem 2 with the package. Are the results similar or different? If there are differences, explain why.

The results are the same, this is because it is based purely on the distance. If there is a difference, it is because of how a kNN decides to resolve a tie which is only applicable to kNN with even values.

**Problem 4 [Cross validation]**

(b) Run the CV experiment for logistic regression and LDA for k ∈ {5, 10, 20}. You can use a package for training the logistic regression and LDA models. Print for each model the average validation error for each value of k.

|  |  |  |
| --- | --- | --- |
| K | Logistic Regression | LDA |
| 5 | 0.9272 | 0.8880 |
| 10 | 0.9280 | 0.8863 |
| 20 | 0.9290 | 0.8861 |

(c) Which model performs better? Compare the results.

Logistic regression performed better where as K increased, the results of the average validation error decreases as well. LDA regresses as K increases although the amount it decreases by is a small amount.