COMP 472 AI Mini-Project 1

AI Tech:
Kevin Rao 40095427
Lydia Fodouop 40132543
Suthan Sinnathurai 40086318
Due October 18th, 2021

TASK 1 ANALYSIS

Analysis of the BBC dataset

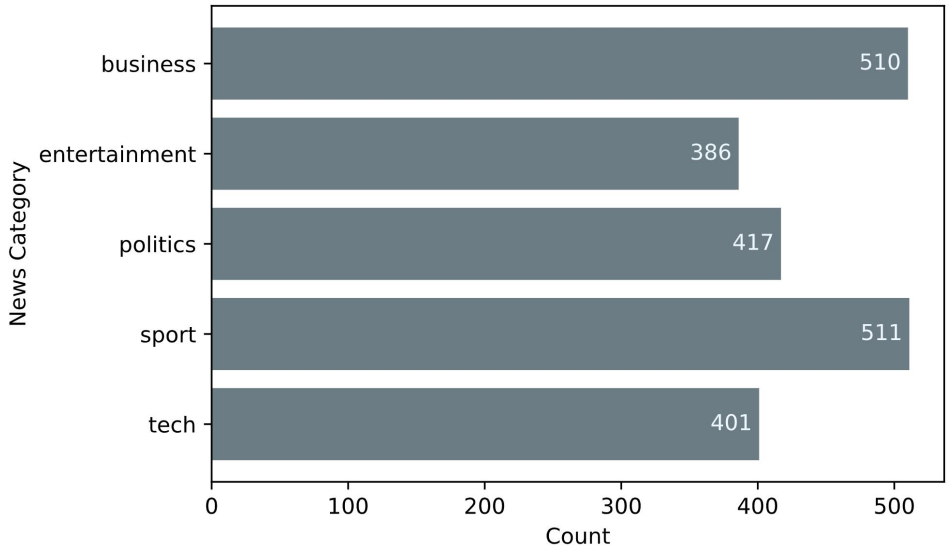
Consists of 2225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005. Natural Classes: 5 (business, entertainment, politics, sport, tech)

-D. Greene and P. Cunningham. "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering", Proc. ICML 2006. (From the readme included with the dataset)

- Overall, the BBC dataset is unbalanced.
  - Business and sport class are represented in more files as opposed to the other classes.
- No specific use case in mind.

The favored metric is weighted F1 score.

Figure 1. BBC-distribution



Result of the Naive Bayes classifier for step 7 & 8

Step 7

```
(a)
*******************************************
**** MultinomialNB default values, try 1 ****
*******************************************
(c)
               precision    recall  f1-score   support


     business       0.99      0.96      0.97       120
entertainment       1.00      0.97      0.99        73
     politics       0.95      0.99      0.97        83
        sport       1.00      1.00      1.00        93
         tech       0.96      1.00      0.98        76
(d)
Accuracy       : 0.9820224719101124
Macro-avg    F1: 0.9823493489542614
Weighted-avg F1: 0.982041965415221
```

Step 8

```
(a)
*******************************************
**** MultinomialNB default values, try 2 ****
*******************************************

(c)
               precision    recall  f1-score   support


     business       0.99      0.96      0.97       120
entertainment       1.00      0.97      0.99        73
     politics       0.95      0.99      0.97        83
        sport       1.00      1.00      1.00        93
         tech       0.96      1.00      0.98        76
(d)
Accuracy       : 0.9820224719101124
Macro-avg    F1: 0.9823493489542614
Weighted-avg F1: 0.982041965415221
```

Result of the Naive Bayes classifier for step 9 & 10

Step 9

(a)

```
*****************************************
**** MultinomialNB with 0.0001 smoothing ****
*****************************************
```

(c)

	precision	recall	f1-score	support
business	0.97	0.96	0.97	120
entertainment	1.00	0.95	0.97	73
politics	0.96	0.96	0.96	83
sport	1.00	1.00	1.00	93
tech	0.93	1.00	0.96	76

(d)

```
Accuracy       : 0.9730337078651685
Macro-avg    F1: 0.9728196557359571
Weighted-avg F1: 0.9730875788249118
```

Step 10

(a)

```
*****************************************
**** MultinomialNB with 0.9 smoothing ****
*****************************************
```

(c)

	precision	recall	f1-score	support
business	0.99	0.96	0.97	120
entertainment	1.00	0.97	0.99	73
politics	0.95	0.99	0.97	83
sport	1.00	1.00	1.00	93
tech	0.96	1.00	0.98	76

(d)

```
Accuracy       : 0.9820224719101124
Macro-avg    F1: 0.9823493489542614
Weighted-avg F1: 0.982041965415221
```

Analysis of the result of the Naive Bayes classifier

```
Step  7: Weighted-avg F1: 0.982041965415221
Step  8: Weighted-avg F1: 0.982041965415221
Step  9: Weighted-avg F1: 0.9730875788249118
Step 10: Weighted-avg F1: 0.982041965415221
```

- Step 7,8,10 share the same performance.
- Step 9 has a slightly worse performance.
- Reason behind the similarity between step 7,8, and 10
  - The same training and test set was used in all models. This will result in all the models having the same priors.
  - Step 8 has the same parameters and hyper-parameters as step 7.
  - Step 10 has smoothing value of 0.9 which is very similar to the smoothing value of both step 7 and 8 that uses 1
- Reason behind the step having a slightly worse performance
  - The smoothing value in step 9 is set at 0.0001 which is different from the smoothing values used in step 7,8 and 10 (1,1 and 0.9).
  - This results in a different conditional probability calculation which results in a slightly worse performance.

Confusion Matrix (CM)

Columns are predicted labels. Rows are true labels.
Ordered alphabetically left-right / top-down.

115 0 3 0 2 0 71 1 0 1 1 0 82 0 0 0 0 0 93 0 0 0 0 0 76 Step 7	115 0 3 0 2 0 71 1 0 1 1 0 82 0 0 0 0 0 93 0 0 0 0 0 76 Step 8
115 0 2 0 3 0 69 1 0 3 3 0 80 0 0 0 0 0 93 0 0 0 0 0 76 Step 9	115 0 3 0 2 0 71 1 0 1 1 0 82 0 0 0 0 0 93 0 0 0 0 0 76 Step 10

- Step 7, 8, 10 share the same CM.
- Step 9 has a different CM.

Reasons to share the same CM:

- All share the same training and testing datasets.
  - Similar calculated Priors and Conditionals.
- Step 8 has same hyper-parameters as step 7.
- Step 10 has very similar smoothing as step 7. (0.9 vs default 1.0)

Reason to differ CM:

- Smoothing for Step 9 is far away from Step 7's. (0.0001 vs default 1.0)

Effect of Smoothing on performance

```
Conditionals formula:
(Word-Frequency-in-Class + Smoothing) ÷
(All-Words-Frequency-in-Class + Smoothing * Vocabulary-Size)
```

Disparity of conditionals (affects score) of different frequency words is greater with 0.0001 smoothing than 1.0 smoothing.

Example Conditionals	Frequency=0	Frequency=20
Smoothing=1.0	`1/(125000 + 27000)` `= 6.6 * 10^-6`	`21/(125000 + 27000)` `= 1.4 * 10^-4`
Smoothing=0.0001	`0.0001/125000+(0.0001*27000)` `= 8.0 * 10^-10`	`20.0001/(125000 + 2.7)` `= 1.6 * 10^-4`

TASK 2 ANALYSIS

Analysis of the Drug200.csv dataset

Drug200.csv dataset was provided for the assignment by Concordia University.

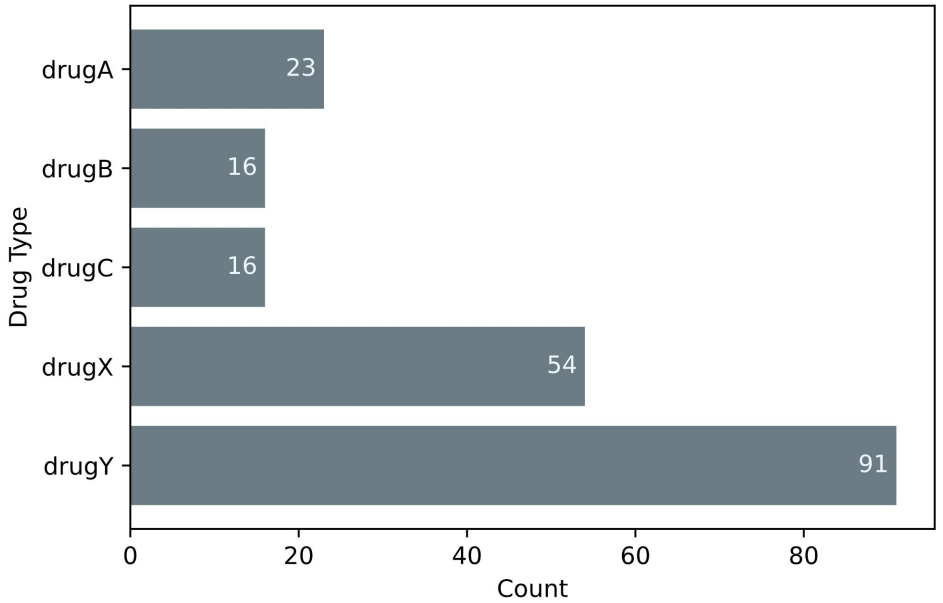
There are 200 entries under 5 classes in total: drugA, drugB, drugC, drugX, drugY.

- Overall, the Drug200 dataset is unbalanced.
  - The drugY class is represented in a lot more instances as opposed to the other classes.
- Use case is providing the proper drug the patient needs.

The favored metric is the weighted recall score.

※vs Use case for precision: conserve drugs, not waste drug.

Figure 2. drug200-distribution



- Binary classifiers, eg Perceptrons, may struggle with multi-labelled data.

Result of the performance of Naive Bayes and Perceptron

```
************
**** NB ****
************
(c)
              precision    recall  f1-score   support

       drugA       0.57      1.00      0.73         4
       drugB       0.67      0.80      0.73         5
       drugC       0.62      1.00      0.77         5
       drugX       1.00      1.00      1.00        12
       drugY       1.00      0.71      0.83        24
(d)
Accuracy       : 0.84
Macro-avg    F1: 0.8106089032918302
Weighted-avg F1: 0.8458809483199726
```

```
************
**** PER ****
************
(c)
              precision    recall  f1-score   support

       drugA       0.00      0.00      0.00         4
       drugB       0.00      0.00      0.00         5
       drugC       0.00      0.00      0.00         5
       drugX       0.31      0.42      0.36        12
       drugY       0.62      0.88      0.72        24
(d)
Accuracy       : 0.52
Macro-avg    F1: 0.21625615763546802
Weighted-avg F1: 0.4333004926108374
```

Result of the performance of the Decision tree and the top-decision tree

(a)

****

**** Base-DT ****

****

(c)

	precision	recall	f1-score	support
drugA	1.00	1.00	1.00	4
drugB	1.00	1.00	1.00	5
drugC	1.00	1.00	1.00	5
drugX	1.00	1.00	1.00	12
drugY	1.00	1.00	1.00	24

(d)

Accuracy : 1.0

Macro-avg F1: 1.0

Weighted-avg F1: 1.0

(a)

****

**** Top-DT ****

**** criterion : gini ****

**** max_depth : None ****

**** min_samples_split: 2 ****

****

(c)

	precision	recall	f1-score	support
drugA	1.00	1.00	1.00	4
drugB	1.00	1.00	1.00	5
drugC	1.00	1.00	1.00	5
drugX	1.00	1.00	1.00	12
drugY	1.00	1.00	1.00	24

(d)

Accuracy : 1.0

Macro-avg F1: 1.0

Weighted-avg F1: 1.0

Result of performance for the base-MLP and the top-MLP

(a)

****

**** Base-MLP ****

****

(c)

	precision	recall	f1-score	support
drugA	0.00	0.00	0.00	4
drugB	0.00	0.00	0.00	5
drugC	0.00	0.00	0.00	5
drugX	0.30	0.50	0.37	12
drugY	0.63	0.79	0.70	24

(d)

Accuracy : 0.5

Macro-avg F1: 0.21574074074074073

Weighted-avg F1: 0.42777777777777787

(a)

****

**** Top-MLP **********

**** activation : tanh ****

**** hidden_layer_sizes: (30, 50) ****

**** solver : adam ****

****

(c)

	precision	recall	f1-score	support
drugA	0.60	0.75	0.67	4
drugB	0.75	0.60	0.67	5
drugC	0.00	0.00	0.00	5
drugX	0.53	0.67	0.59	12
drugY	0.81	0.88	0.84	24

(d)

Accuracy : 0.7

Macro-avg F1: 0.5531851851851851

Weighted-avg F1: 0.6654222222222222

Analysis of the performance of all 6 classifiers (Step 7)

Gaussian NB weighted recall	≈0.84
Decision tree weighted recall	≈1.0
Top Decision tree weighted recall	≈1.0
Perceptron weighted recall	≈0.52
MLP weighted recall	≈0.50
Top MLP weighted recall	≈0.70

The model with highest weighted recall are both the base and top decision trees with a recall of 100 %.

The model with the lowest weighted recall was calculated to be the MLP (50%).

Ordered by performance:
Top-DT = Base-DT > NB > Top-MLP > PER > Base-MLP

Analysis of the performance stability of Gaussian NB, Decision tree, top-decision tree and Perceptron (Step 8)

```
NB:
        Accuracy       :
                mean: 0.84
                pstd: 0.0
        Macro-avg    F1:
                mean: 0.8106089032918302
                pstd: 0.0
        Weighted-avg F1:
                mean: 0.8458809483199726
                pstd: 0.0
```

```
PER:
        Accuracy       :
                mean: 0.52
                pstd: 0.0
        Macro-avg    F1:
                mean: 0.21625615763546802
                pstd: 0.0
        Weighted-avg F1:
                mean: 0.4333004926108374
                pstd: 0.0
```

```
Base-DT:
        Accuracy       :
                mean: 1.0
                pstd: 0.0
        Macro-avg    F1:
                mean: 1.0
                pstd: 0.0
        Weighted-avg F1:
                mean: 1.0
                pstd: 0.0
```

```
Top-DT:
        Accuracy       :
                mean: 1.0
                pstd: 0.0
        Macro-avg    F1:
                mean: 1.0
                pstd: 0.0
        Weighted-avg F1:
                mean: 1.0
                pstd: 0.0
```

- All four classifiers here have a stdev. of 0.0 for all the metrics.
- The performance of each of those models did not change between each of the 10 runs.
  - Each prior and conditional probability remains the same given the same input 10 times resulting in the same performance across runs for the naive Bayes.
  - For the decision trees, due to the same inputs for the test set, we get the same information gain every time resulting in the same ranking of features and thus the same performance every time.
  - The initial weight of the perceptron was set at one for each of the features. For each of the 10 runs, the models set the same initial weights and adjust the weights in the same manner resulting in the same final weighs of the features and performance in the test set.

Analysis of the performance stability of the Base-MLP and the top-MLP (Step 8)

```
Base-MLP:
        Accuracy       :
                mean: 0.5
                pstd: 0.0
        Macro-avg    F1:
                mean: 0.21572678396871944
                pstd: 4.187031606385227e-05
        Weighted-avg F1:
                mean: 0.4274539589442816
                pstd: 0.0009714565004887932

Top-MLP:
        Accuracy       :
                mean: 0.81
                pstd: 0.03492849839314594
        Macro-avg    F1:
                mean: 0.7187314959170295
                pstd: 0.07514048795758249
        Weighted-avg F1:
                mean: 0.7886299723506699
                pstd: 0.0468162497129567
```

```
*****************************************
**** B-MLP (10 runs)   ****
*****************************************


Average Accuracy (10 runs)  : 0.58
Average Macro F1(10 runs): 0.58
Average Weighted F1 (10 runs): 0.4818181818181818

Average Accuracy Stdev: 0.018973665961010237
Average MacroF1 Stdev: 0.011900083986743052
Average Weighted Stdev: 0.02035856096882395
```

- Both MLP models have a non-zero standard deviation for the macro F1 and the weighted F1 score.
- The accuracy of the base MLP was calculated to have a stdev. of 0.0 for the first 10 runs .
- The stdev. of the accuracy was non-zero when we ran the model 10 times again.
- This means that the performance of each of those models changed between each of the 10 runs.
    - For the MLP models, the weights and the bias are initialized randomly (Source: Sklearn documentation on MLP Classifier's random_state)
    - This will result in a different performance output every time the model is trained and tested.

Confusion Matrix Task 2

Columns are predicted labels. Rows are true labels.
Ordered alphabetically left-right / top-down.

4 0 0 0 0 1 4 0 0 0 0 0 5 0 0 0 0 0 12 0 2 2 3 0 17 NB	4 0 0 0 0 0 5 0 0 0 0 0 5 0 0 0 0 0 12 0 0 0 0 0 24 Base-DT	4 0 0 0 0 0 5 0 0 0 0 0 5 0 0 0 0 0 12 0 0 0 0 0 24 Top-DT
0 0 0 2 2 0 0 0 5 0 0 0 0 1 4 0 0 0 5 7 0 0 0 3 21 PER	0 0 0 2 2 0 0 0 5 0 0 0 0 2 3 0 0 0 6 6 0 0 0 5 19 Base-MLP	3 1 0 0 0 0 3 0 2 0 0 0 0 2 3 2 0 0 8 2 0 0 0 3 21 Top-MLP

Team Contributions

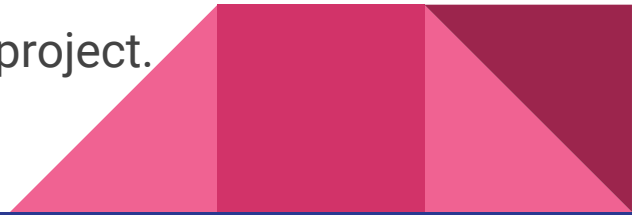
Contributions

Strategy for the Mini-project we agreed upon:

1. Everyone did each task independently.
2. On team meetings, we compared each other's work (methodologies/code) and output.
3. Combine the best part of everyone's code for the submission.

Also, we all discussed the results together,
and built the presentation slides together.

… So everyone has worked on all the aspects of the mini-project.



Contributions

In practice:

- Everyone’s code ended up being very similar to each other’s.
  - The same calls to sklearn’s module were done in roughly the same order.
- The tasks took more effort than expected
  - We were pressed for time.

So we took the shortest code, and added notable improvements from the others onto it.