Analysis of Different ML Models

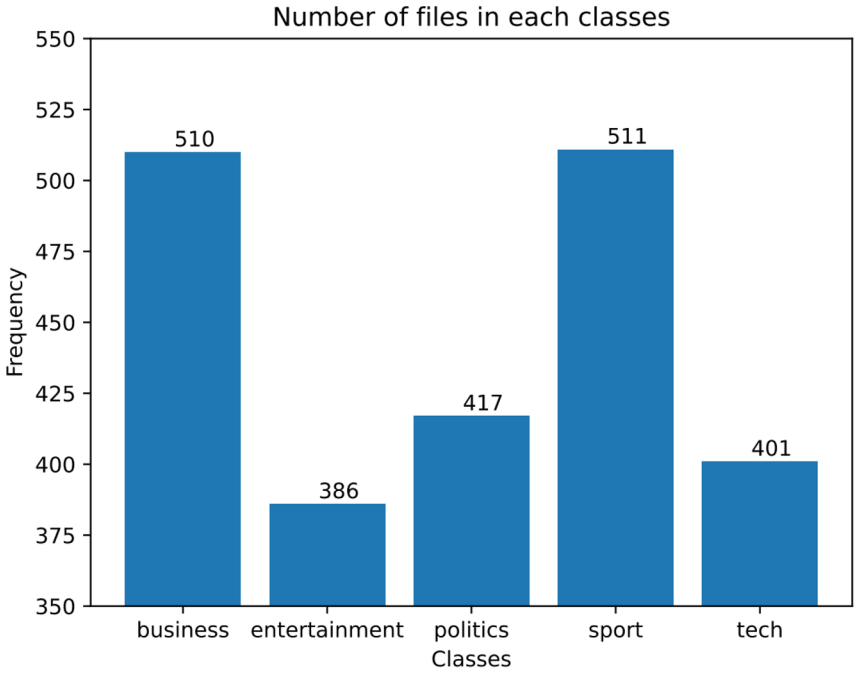
COMP472 Assignment 1

Contributor: Rhina Kim - 40130779

Task 1: Text Classification (Initial Dataset)

Characteristics of initial dataset

- There are 5 classes: **Business, Entertainment, Politics, Sport, and Tech**
- **Imbalanced datasets** (distribution not uniform)
- Multinomial Naive Bayes does not perform well on imbalanced datasets → **May overfit data** in favor of the class with more number of examples (Testing error may be high)
- Can affect metrics system (F1 measures are superior than accuracy



Task 1: Text Classification (Results)

1. Multinomial default values: **Try 1 and Try 2**

Classifier:
MultinomialNB(**smoothing=1.0**)

Result:
- Same results for both of the trials
- High Accuracy (98.2%)
- High f1 macro avg (98.14%)
- High f1 weighted avg (98.21%)

Reason:
- No noisy-inputs
- Small features with low-frequency

```
(b) Confusion matrix:
[[113   2   1   0   2]
 [  0  70   0   0   2]
 [  0   0  83   0   0]
 [  0   0   0  93   0]
 [  0   1   0   0  78]]
```

(c) Classification report (Precision, Recall, F1):

Classes		precision	recall	f1-score	support
business	0	1.00	0.96	0.98	118
entertainment	1	0.96	0.97	0.97	72
politics	2	0.99	1.00	0.99	83
sport	3	1.00	1.00	1.00	93
tech	4	0.95	0.99	0.97	79
accuracy				0.98	445
macro avg		0.98	0.98	0.98	445
weighted avg		0.98	0.98	0.98	445

Task 1: Text Classification (Results)

1. Multinomial with **smoothing value = 0.0001**

```
(b) Confusion matrix:
[[113   1   1   0   3]
 [  0  70   0   0   2]
 [  0   1  82   0   0]
 [  0   0   0  93   0]
 [  0   1   1   0  77]]
```

Classifier:
MultinomialNB(**smoothing=0.0001**)

Result:
- High Accuracy (97.75%)
- High f1 macro avg (97.65%)
- High f1 weighted avg (97.76%)

Reason:
- Result is lower than default smoothing value
  → the smoothed values are closer to the actual data (**Higher possibility of zero probability**)

(c) Classification report (Precision, Recall, F1):

Classes		precision	recall	f1-score	support
business	0	1.00	0.96	0.98	118
entertainment	1	0.96	0.97	0.97	72
politics	2	0.98	0.99	0.98	83
sport	3	1.00	1.00	1.00	93
tech	4	0.94	0.97	0.96	79
accuracy				0.98	445
macro avg		0.97	0.98	0.98	445
weighted avg		0.98	0.98	0.98	445

Task 1: Text Classification (Results)

1. Multinomial with **smoothing value = 0.9**

Classifier:
MultinomialNB(**smoothing=0.9**)

Result:
- Same results for Trial 1 & 2
- High Accuracy (98.2%)
- High f1 macro avg (98.14%)
- High f1 weighted avg (98.21%)

Reason:
- Smoothing value is very close to default smoothing value (1.0)
  → Differences in result not significant as first trials

(b) Confusion matrix:

```
[[113   2   1   0   2]
 [  0  70   0   0   2]
 [  0   0  83   0   0]
 [  0   0   0  93   0]
 [  0   1   0   0  78]]
```

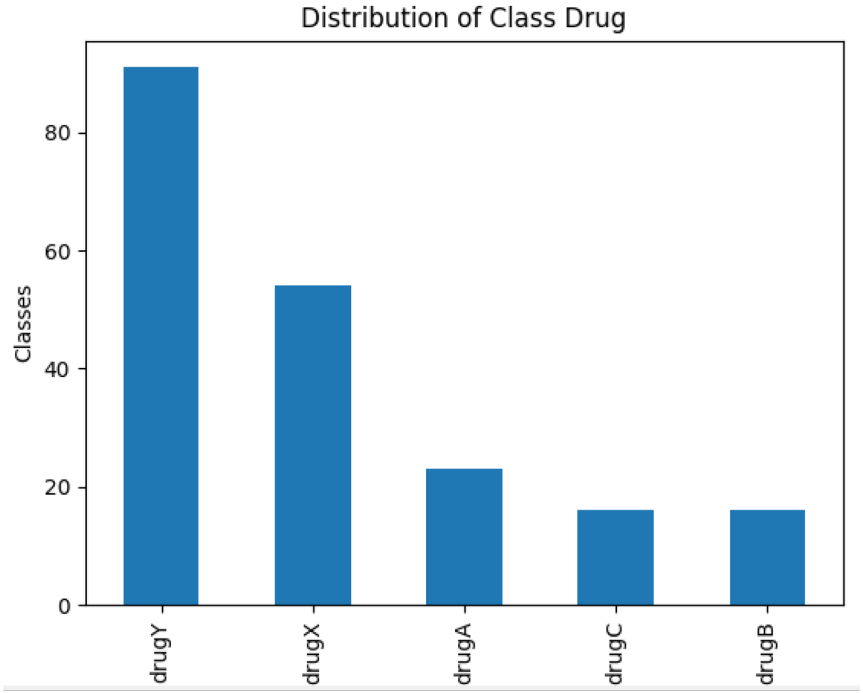
(c) Classification report (Precision, Recall, F1):

Classes		precision	recall	f1-score	support
business	0	1.00	0.96	0.98	118
entertainment	1	0.96	0.97	0.97	72
politics	2	0.99	1.00	0.99	83
sport	3	1.00	1.00	1.00	93
tech	4	0.95	0.99	0.97	79
accuracy				0.98	445
macro avg		0.98	0.98	0.98	445
weighted avg		0.98	0.98	0.98	445

Task 2: Drug Classification (Initial Dataset)

Characteristics of initial dataset

- There are 5 classes: **drugY, drugX, drugA, drugC, drugB**
- **Imbalanced datasets** (distribution not uniform)
- Most of the classifiers does not perform well on imbalanced datasets
  → **May overfit data** in favor of the class with more number of examples (Testing error may be high)
- Can affect metrics system (F1 measures are superior than accuracy)



Task 2: Drug Classification (Results)

1. Classifier: **Gaussian(NB)**
   with Default smoothing value

Result:
- Moderate to High Accuracy (77.5%)
- Moderate to High f1 macro avg (71.89%)
- Moderate to High f1 weighted avg (78.12%)

Reason:
- Initial Dataset was unbalanced
- Recall higher than precision for most of the classes
  → A lot of predicted instances are labeled correctly
- drugX score higher since it

(b) Confusion matrix:

```
[[ 3  0  0  0  0]
 [ 0  3  0  0  0]
 [ 0  0  2  0  0]
 [ 0  0  0 13  0]
 [ 3  3  3  0 10]]
```

(c) Classification report (Precision, Recall, F1):

Classes		precision	recall	f1-score	support
drugA	0	0.50	1.00	0.67	3
drugB	1	0.50	1.00	0.67	3
drugC	2	0.40	1.00	0.57	2
drugX	3	1.00	1.00	1.00	13
drugY	4	1.00	0.53	0.69	19
accuracy				0.78	40
macro avg		0.68	0.91	0.72	40
weighted avg		0.89	0.78	0.78	40

Task 2: Drug Classification (Results)

```
[[ 3  0  0  0  0]
 [ 0  3  0  0  0]
 [ 0  0  2  0  0]
 [ 0  0  0 13  0]
 [ 0  0  0  0 19]]
```

1. Classifier: Default
   DecisionTreeClassifier()
   Criterion: gini
   Min-split-sample: 2
   Max-depth: None

Result:
- Highest Accuracy (100%)
- Highest f1 macro avg (100%)
- Highest f1 weighted avg (100%)

Reason:
- Due to its ability to use different feature subsets and decision rules
  → Best for categorical data
- Easy to handle categorical and numerical variables

(c) Classification report (Precision, Recall, F1):

Classes	precision	recall	f1-score	support
0	1.00	1.00	1.00	3
1	1.00	1.00	1.00	3
2	1.00	1.00	1.00	2
3	1.00	1.00	1.00	13
4	1.00	1.00	1.00	19
accuracy			1.00	40
macro avg	1.00	1.00	1.00	40
weighted avg	1.00	1.00	1.00	40

Task 2: Drug Classification (Results)

1. Classifier: Best
   DecisionTreeClassifier()
   criterion: gini
   max_depth: 6
   min_samples_leaf: 0.02

Result: Same as default decision tree classifier

Reason: Same as default decision tree classifier
- Range of Entropy : 0 to 1
- Range of Gini Impurity: 0 to 0.5
  → Gini Impurity is better since entropy range is small to select the best features

```
[[ 3  0  0  0  0]
 [ 0  3  0  0  0]
 [ 0  0  2  0  0]
 [ 0  0  0 13  0]
 [ 0  0  0  0 19]]
```

(c) Classification report (Precision, Recall, F1):

Classes	precision	recall	f1-score	support
0	1.00	1.00	1.00	3
1	1.00	1.00	1.00	3
2	1.00	1.00	1.00	2
3	1.00	1.00	1.00	13
4	1.00	1.00	1.00	19
accuracy			1.00	40
macro avg	1.00	1.00	1.00	40
weighted avg	1.00	1.00	1.00	40

Task 2: Drug Classification (Results)

1. Classifier: Default
   Perceptron()

```
(b) Confusion matrix:
[[ 0  0  0  3  0]
 [ 0  0  0  3  0]
 [ 0  0  0  2  0]
 [ 0  0  0 12  1]
 [ 0  0  0  9 10]]
```

Result:
- Low Accuracy (55%)
- Low macro avg (24.76%)
- Low weighted avg (50.24%)

Reason:
- Perceptron was not really suitable for categorical dataset with mainly nominal attributes
  → perceptron is mostly used to classify binary data

(c) Classification report (Precision, Recall, F1):

Classes	precision	recall	f1-score	support
0	0.00	0.00	0.00	3
1	0.00	0.00	0.00	3
2	0.00	0.00	0.00	2
3	0.41	0.92	0.57	13
4	0.91	0.53	0.67	19
accuracy			0.55	40
macro avg	0.26	0.29	0.25	40
weighted avg	0.57	0.55	0.50	40

Task 2: Drug Classification (Results)

1. Classifier: Default
   MLPClassifier()
   hidden_layer_sizes: 100
   activation: logistic f(x) = 1 / (1 + exp(-x))
   solver: sdg

Result:
- Low Accuracy (65%)
- Low macro avg (27.85%)
- Low weighted avg (57.26%)

Reason:

(b) Confusion matrix:

```
[[ 0  0  0  1  2]
 [ 0  0  0  3  0]
 [ 0  0  0  1  1]
 [ 0  0  0  8  5]
 [ 0  0  0  1 18]]
```

(c) Classification report (Precision, Recall, F1):

Classes	precision	recall	f1-score	support
0	0.00	0.00	0.00	3
1	0.00	0.00	0.00	3
2	0.00	0.00	0.00	2
3	0.57	0.62	0.59	13
4	0.69	0.95	0.80	19
accuracy			0.65	40
macro avg	0.25	0.31	0.28	40
weighted avg	0.51	0.65	0.57	40

Task 2: Drug Classification (Results)

1. Classifier: Best
   MLPClassifier()
   hidden_layer_sizes: (30, 50)
   activation: tanh
   solver: adam

Result:
- Low Accuracy (95%)
- Low macro avg (91.83%)
- Low weighted avg (94.91%)

Reason:

(b) Confusion matrix:

```
[[ 2  1  0  0  0]
 [ 0  3  0  0  0]
 [ 0  0  2  0  0]
 [ 0  0  0 12  1]
 [ 0  0  0  0 19]]
```

(c) Classification report (Precision, Recall, F1):

Classes	precision	recall	f1-score	support
0	1.00	0.67	0.80	3
1	0.75	1.00	0.86	3
2	1.00	1.00	1.00	2
3	1.00	0.92	0.96	13
4	0.95	1.00	0.97	19
accuracy			0.95	40
macro avg	0.94	0.92	0.92	40
weighted avg	0.96	0.95	0.95	40