Lab7

May 17, 2023

0.1 1.1. Poisonous Mushrooms

	Predicted (M1)		
	Poisonous	Edible	Total
Actual	3	0	Negative: 3
	3	4	Positive: 7
	6	4	Total: 10

	Predicted (M2)		
	Poisonous	Edible	Total
Actual	2	1	Negative: 3
	0	7	Positive: 7
	2	8	Total: 10

1.1.1. Show the confusion matrices for models M1 and M2.

1.1.2. Compute accuracy, precision, recall for both models. Accuracy = (TP + TN)/(TP+TN+FP+FN)
Precision = TP/(TP+FP)
Recall = TP/(TP+FN)

[1]:
```
acc_m1 = (4 + 3)/(4 + 3 + 0 + 3)
pre_m1 = 4/(4 + 0)
rec_m1 = 4/(4 + 3)

print("Model 1")
print(f"Accuracy: {acc_m1}")
print(f"Precision: {pre_m1}")
print(f"Recall: {rec_m1}")

acc_m2 = (7 + 2)/(7 + 2 + 1 + 0)
pre_m2 = 7/(7 + 1)
rec_m2 = 7/(7 + 0)

print("\nModel 2")
```

```
print(f"Accuracy: {acc_m2}")
print(f"Precision: {pre_m2}")
print(f"Recall: {rec_m2}")
```

```
Model 1
Accuracy: 0.7
Precision: 1.0
Recall: 0.5714285714285714

Model 2
Accuracy: 0.9
Precision: 0.875
Recall: 1.0
```

3. Prof. Joffe wants to get the app out tomorrow. Which model, M1 or M2 will you recommend him to use? Explain your reasoning! I would recommend Model 2, which has a higher accuracy, which means 90% of the predictions turned out to be correct. It has a slightly lower precision, which is how much of the predicted positive turned out to be correct, which wouldn't be too much of a worry because model 2 has a higher accuracy. Finally Model 2 has a higher recall, which meant that it got 100% of the actual positives correct.

0.2 1.2. Defendants

	Predicted (M1)		
	Guilty	Innocent	Total
Actual	4	1	Negative: 5
	2	3	Positive: 5
	6	4	Total: 10

	Predicted (M2)		
	Guilty	Innocent	Total
Actual	2	3	Negative: 5
	0	5	Positive: 5
	2	8	Total: 10

1.2.1 Show the confusion matrices for M3 and M4.

1.2.2. Compute accuracy, precision, recall for both models. Accuracy = (TP + TN)/(TP+TN+FP+FN)
Precision = TP/(TP+FP)
Recall = TP/(TP+FN)

[2]:
```
acc_m1 = (3 + 4)/(10)
pre_m1 = 3/(3 + 1)
```

```
rec_m1 = 3/(3 + 2)

print("Model 1")
print(f"Accuracy: {acc_m1}")
print(f"Precision: {pre_m1}")
print(f"Recall: {rec_m1}")

acc_m2 = (5 + 2)/(10)
pre_m2 = 5/(5 + 3)
rec_m2 = 5/(5 + 0)

print("\nModel 2")
print(f"Accuracy: {acc_m2}")
print(f"Precision: {pre_m2}")
print(f"Recall: {rec_m2}")
```

```
Model 1
Accuracy: 0.7
Precision: 0.75
Recall: 0.6

Model 2
Accuracy: 0.7
Precision: 0.625
Recall: 1.0
```

1.2.3. Cole-Tindall wants to commission the AI system tomorrow. Which model would you recommend her to use? Explain your reasoning! I would use model 2, because it has a the same accuracy, which means they resulted in the same percentage 70% of predicting the results right. It has a lower precision, which means not all innocents it predicted turned out to be innocent. But it also has a higher recall, which meant it predicted all of the actuall innocent correctly.

0.3 1.3. Defendants again

	Predicted (M1)		
	Innocent	Guilty	Total
Actual	3	2	Negative: 5
	1	4	Positive: 5
	3	6	Total: 10

	Predicted (M2)		
	Innocent	Guilty	Total
Actual	5	0	Negative: 5
	3	2	Positive: 5

	Predicted (M2)		
	8	2	Total: 10

1.3.1. Construct the corresponding confusion matrices

1.3.2. Compute accuracy, precision, recall for both models

Accuracy = (TP + TN)/(TP+TN+FP+FN)
Precision = TP/(TP+FP)
Recall = TP/(TP+FN)

[3]:
```
acc_m1 = (4 + 3)/(10)
pre_m1 = 4 /(4 + 2)
rec_m1 = 4/(4 + 1)

print("Model 1")
print(f"Accuracy: {acc_m1}")
print(f"Precision: {pre_m1}")
print(f"Recall: {rec_m1}")

acc_m2 = (2 + 5)/(10)
pre_m2 = 2/(2 + 0)
rec_m2 = 2/(2 + 3)

print("\nModel 2")
print(f"Accuracy: {acc_m2}")
print(f"Precision: {pre_m2}")
print(f"Recall: {rec_m2}")
```

```
Model 1
Accuracy: 0.7
Precision: 0.6666666666666666
Recall: 0.8

Model 2
Accuracy: 0.7
Precision: 1.0
Recall: 0.4
```

3. Explain the differences between this and the previous approach: what difference does it make if you swap around which class is taken as positive? The recall for model 2 dropped a lot, while the precision rose. The recall of model 1 rose while precision dropped. By swapping the positive class, we gain more information about which class the model is better at describing.

[ ]:
```

```