COMP47750/COMP47990 Tutorial

Evaluation in Machine Learning

1. The *confusion matrix* below shows the evaluation results for a binary classifier when applied to a test set of 768 examples, which are annotated with the class labels: (Pass, Fail).

	Predicted Class: Fail	Predicted Class: Pass	
Real Class	160	108	Fail
Real Class	93	407	Pass

From this table calculate:

a) The *precision* score for both of the classes.

b) The *recall* score for both of the classes.

c) The *F1-measure* score for both of the classes.

d) The *overall classification accuracy* for the full test set.

2. The table below shows the true class labels for a test set of 12 emails, which are labelled as “spam” or “non-spam”. The table also reports the predictions made by three different binary classifiers for those emails.

Example	True Class Label	KNN Prediction	D-Tree Prediction	SVM Prediction
1	spam	spam	spam	spam
2	non-spam	non-spam	spam	non-spam
3	spam	non-spam	non-spam	spam
4	non-spam	non-spam	non-spam	non-spam
5	spam	spam	spam	spam
6	non-spam	non-spam	non-spam	non-spam
7	non-spam	spam	spam	non-spam
8	non-spam	non-spam	spam	spam
9	spam	spam	non-spam	spam
10	spam	spam	non-spam	non-spam
11	spam	non-spam	non-spam	spam
12	spam	spam	spam	spam

a) Calculate the *overall accuracy* for each of the classifiers on this data. Based on your calculations, which classifier is the most accurate?

b) Calculate the *precision* of each classifier relative to the “spam” class. Based on your calculations, which classifier achieves the highest precision for this class?

3. The table below shows the number of correct and incorrect predictions made by an image classifier during a 10-fold cross validation experiment, where the goal was to classify 5,000 images into one of three categories: {cats, dogs, people} (i.e. each test set contains 500 images).

Fold	**Class: Cats**		**Class: Dogs**		**Class: People**	
	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
1	82	68	82	68	164	36
2	81	69	102	48	176	24
3	99	51	97	53	160	40
4	81	69	102	48	148	52
5	94	56	99	51	148	52
6	97	53	91	59	162	38
7	81	69	94	56	148	52
8	76	74	79	71	181	19
9	76	74	97	53	160	40
10	96	54	79	71	179	21

a) What is the *overall accuracy* of the classifier based on the cross-validation results?

b) What conclusion might be drawn about the different classes in the data, based on the results above?

c) Would *leave-one-out cross validation* be an appropriate evaluation strategy on this dataset?

4. The code below is available in the notebook **09 Evaluation Tutorial**.

   This code will create a synthetic dataset with 50 samples.

```
from sklearn.datasets import make_classification
X, y = make_classification(n_samples=50, n_features=4, n_classes=2
                           class_sep=0.55, random_state=2)
```

   Divide this dataset into 50% train and 50% test and test the performance of a Gaussian Naive Bayes classifier on the data. Repeat this test 10 times and find the mean and standard deviation of the results (code provided).

   Increase the dataset to 1000 samples (`n_samples=1000`) and retest. What happens to the mean and standard deviation? What do we learn from this?

5. The notebook **09 Evaluation Tutorial** contains code for loading the diabetes dataset (**diabetes.csv**).

   a) Using the code in **08 ROC** as a template, produce ROC curves for kNN and Naive Bayes classifiers on the diabetes data.

   b) Repeat this exercise using synthetic data generated using the code below. What insights do these ROC curves provide?

```
from sklearn.datasets import make_classification
X, y = make_classification(n_samples=1000, n_features=4, n_classes=2
                           class_sep=0.75, random_state=1)
```