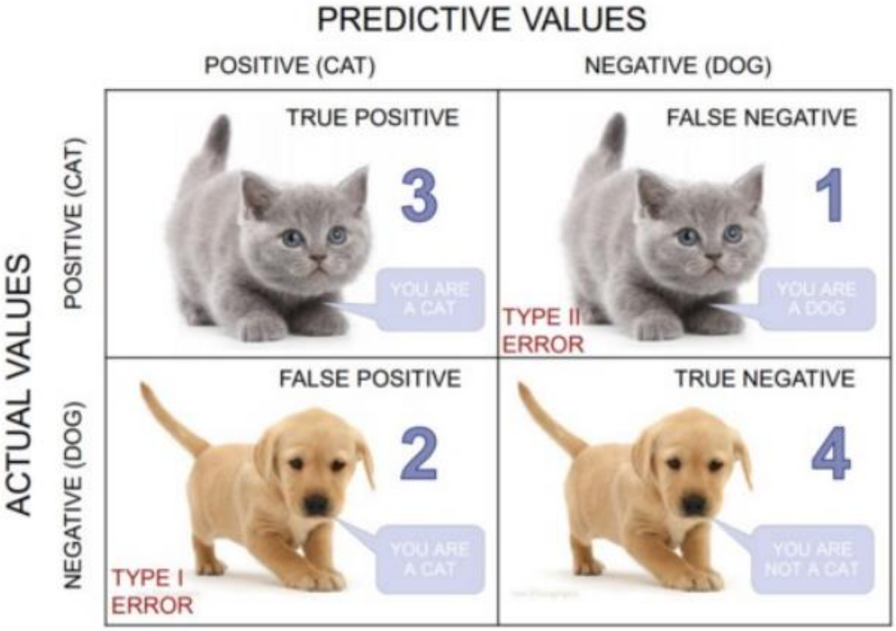
Assignment 2 Part 1

1. What is the role of a *confusion matrix* in the evaluation of a machine trained for a pattern recognition task? In your answer, refer to a concrete example, either from literature or one you created. Anchor you answer in relevant literature.

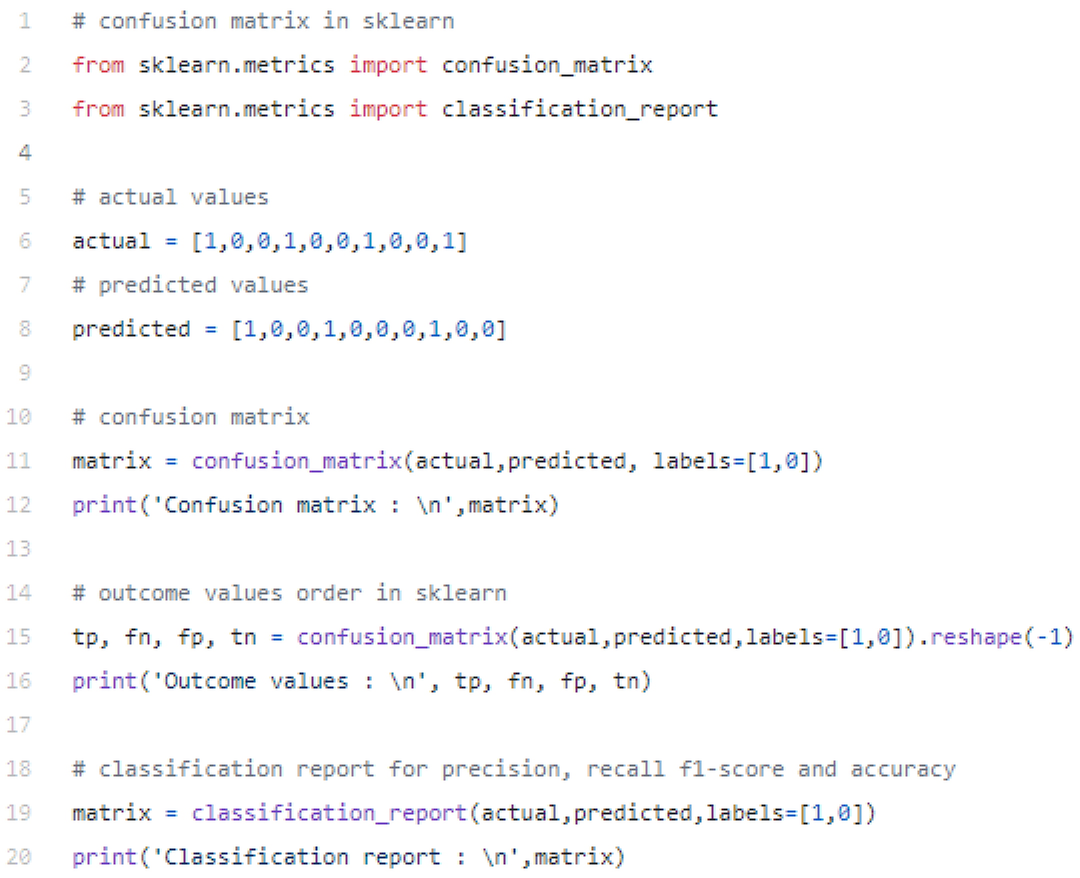
A confusion matrix better conveys accuracy/errors of a machine learning model, as it summarizes the correct and incorrect predictions broken down by each class of data in the dataset, as well as what types of errors are being made. In this way, it displays when the model is confused making predictions.

This image gives examples of how the confusion matrix should be interpreted:

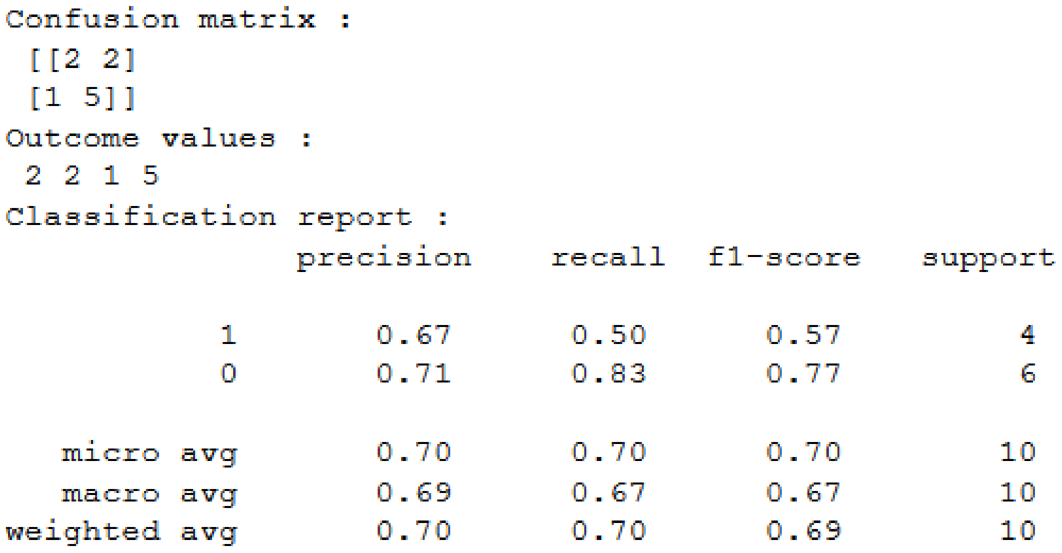


such that a false positive is a type 1 error and false negative is a type 2 error (Sharma, 2019).

Using the sklearn package in Python, a confusion matrix and a classification report can be generated, as seen below:



```python
# confusion matrix in sklearn
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

# actual values
actual = [1,0,0,1,0,0,1,0,0,1]
# predicted values
predicted = [1,0,0,1,0,0,0,1,0,0]

# confusion matrix
matrix = confusion_matrix(actual,predicted, labels=[1,0])
print('Confusion matrix : \n',matrix)

# outcome values order in sklearn
tp, fn, fp, tn = confusion_matrix(actual,predicted,labels=[1,0]).reshape(-1)
print('Outcome values : \n', tp, fn, fp, tn)

# classification report for precision, recall f1-score and accuracy
matrix = classification_report(actual,predicted,labels=[1,0])
print('Classification report : \n',matrix)
```

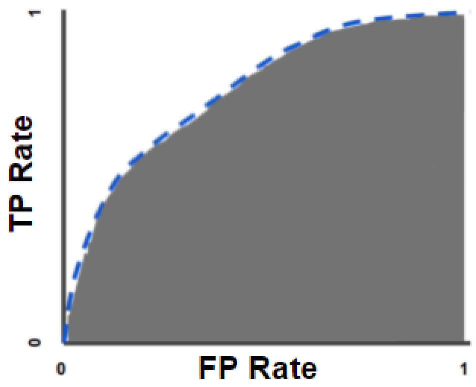


```
Confusion matrix :
 [[2 2]
 [1 5]]
Outcome values :
 2 2 1 5
Classification report :
               precision    recall  f1-score   support

           1       0.67      0.50      0.57         4
           0       0.71      0.83      0.77         6

   micro avg       0.70      0.70      0.70        10
   macro avg       0.69      0.67      0.67        10
weighted avg       0.70      0.70      0.69        10
```

(Bhandari, 2020b).

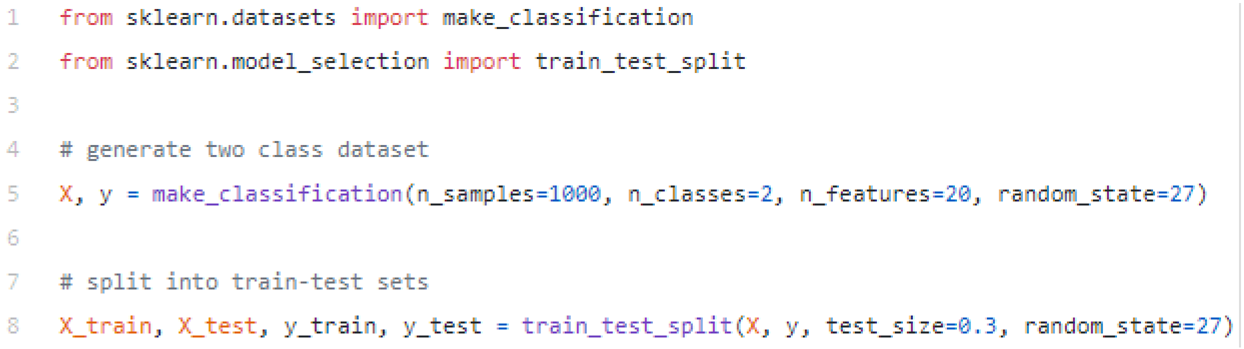
2. What is the role of the ROC curve? How would you use it to compare the performance of several classifiers? In your answer, refer to concrete examples of classifiers, either from literature or one you created. Illustrate the ROC curves and anchor your answer in relevant literature.

A receiver operating characteristic (ROC) curve plots the true positive and false positive rates of a classification model. To compare multiple classifiers, we can find the area under the ROC curve (AUC). The AUC can be interpreted as the probability that the model ranks a random positive example more highly than a random negative example (Classification, n.d.).

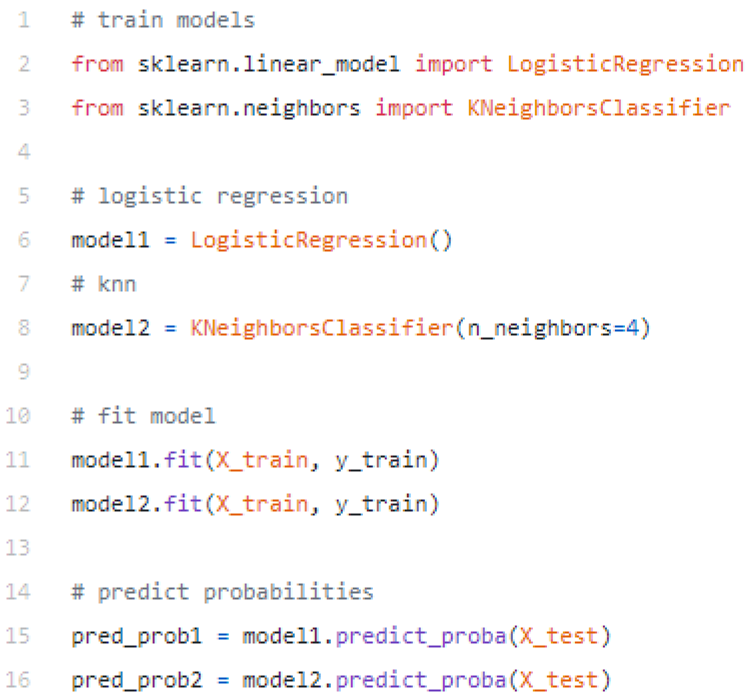


(Classification, n.d.)

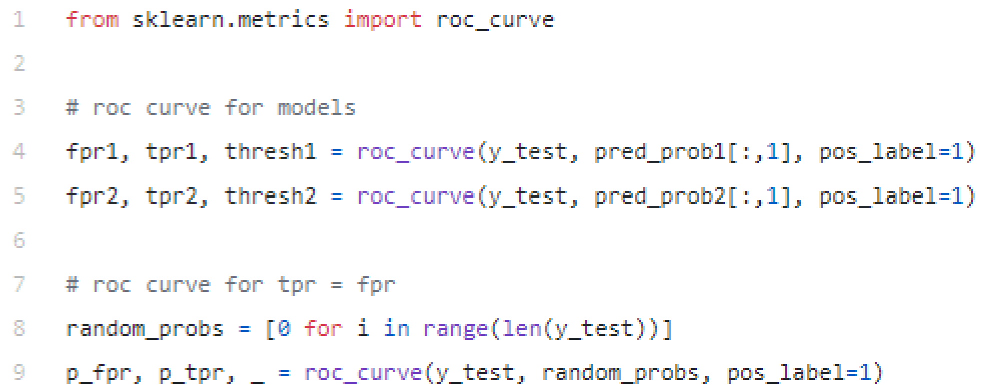
An example of calculating the ROC curve and AUC in Python with randomly generated data can be seen below:



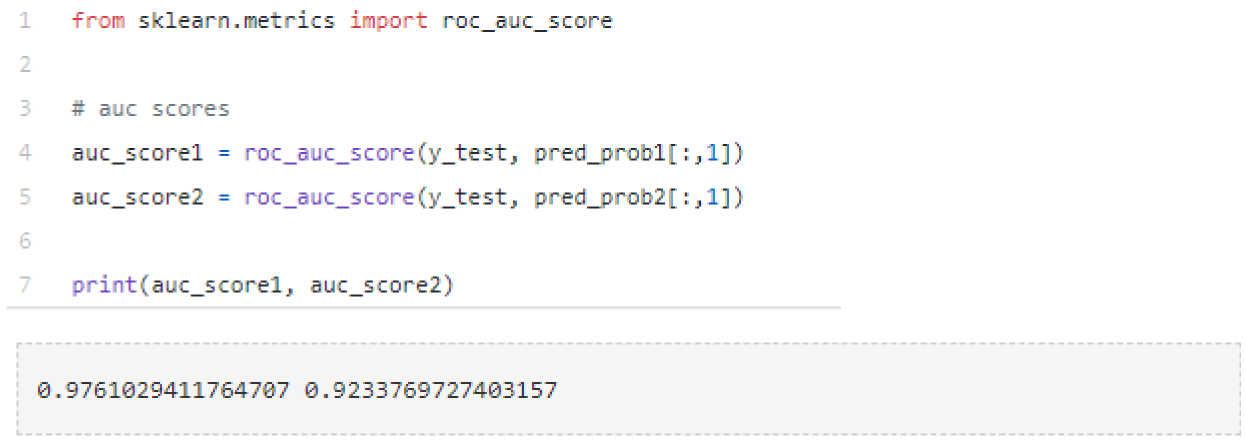
```
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split

# generate two class dataset
X, y = make_classification(n_samples=1000, n_classes=2, n_features=20, random_state=27)

# split into train-test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=27)
```



```python
# train models
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier

# logistic regression
model1 = LogisticRegression()
# knn
model2 = KNeighborsClassifier(n_neighbors=4)

# fit model
model1.fit(X_train, y_train)
model2.fit(X_train, y_train)

# predict probabilities
pred_prob1 = model1.predict_proba(X_test)
pred_prob2 = model2.predict_proba(X_test)
```



```python
from sklearn.metrics import roc_curve

# roc curve for models
fpr1, tpr1, thresh1 = roc_curve(y_test, pred_prob1[:,1], pos_label=1)
fpr2, tpr2, thresh2 = roc_curve(y_test, pred_prob2[:,1], pos_label=1)

# roc curve for tpr = fpr
random_probs = [0 for i in range(len(y_test))]
p_fpr, p_tpr, _ = roc_curve(y_test, random_probs, pos_label=1)
```



```python
from sklearn.metrics import roc_auc_score

# auc scores
auc_score1 = roc_auc_score(y_test, pred_prob1[:,1])
auc_score2 = roc_auc_score(y_test, pred_prob2[:,1])

print(auc_score1, auc_score2)
```

```
0.9761029411764707 0.9233769727403157
```

(Bhandari, 2020a).

Resources:

Bhandari, A. (2021, June 16). AUC-ROC Curve in Machine Learning Clearly Explaining. https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

Bhandari, A. (2020, April 17). *Everything you Should Know about Confusion Matrix for Machine Learning*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/

Classification: ROC Curve and AUC. (n.d.). *Machine Learning Crash Course*. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Sharma, P. (2019, July 21). *Decoding the Confusion Matrix*. Towards data science. https://towardsdatascience.com/decoding-the-confusion-matrix-bb4801decbb