3) Classification

- "Error rate" is basically 1 - `sklearn.metrics.accuracy_score`
- StratifiedShuffleSplit: It basically just ensures that the distribution of different classes is the same in the training nd test se. So for eg, suppose our dataset has 20% data of class1 and 80% of class2, stratified shuffle split ill ensure that our training and test set will both also have 20% data of class1 and 80% data of class2

Error Metrics

Accuracy is not a good metric for classification, especially binary classification. For example, suppose we have a dataset where 99% of patients are healthy and 1% have cancer. Even a useless model which would predict "healthy" all the time will get a 99% accuracy. Choosing the right approach depends on the use case

- **ROC Curve:** Generally better for data with balanced classes
- **Precision-Recall Curve:** Generally better for data with imbalanced classes

Binary Class

Confusion Matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

FP is also called a type I error and FN is called a type II error

In here, accuracy = (TP+TN) / (TP+FN+FP+TN) (ie all n of all our samples)

This is the most popular error metric but isn't good for the example we gave above, ie cases where the population is skewed

Also, recall (aka sensitivity) = TP / (TP+FN)

And precision = TP / (TP+FP)

Specificity = TN / (TN+FP)

F1 (aka harmonic mean) = 2 * [(Precision*Recall) / (Precision+Recall)]

$$\textbf{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

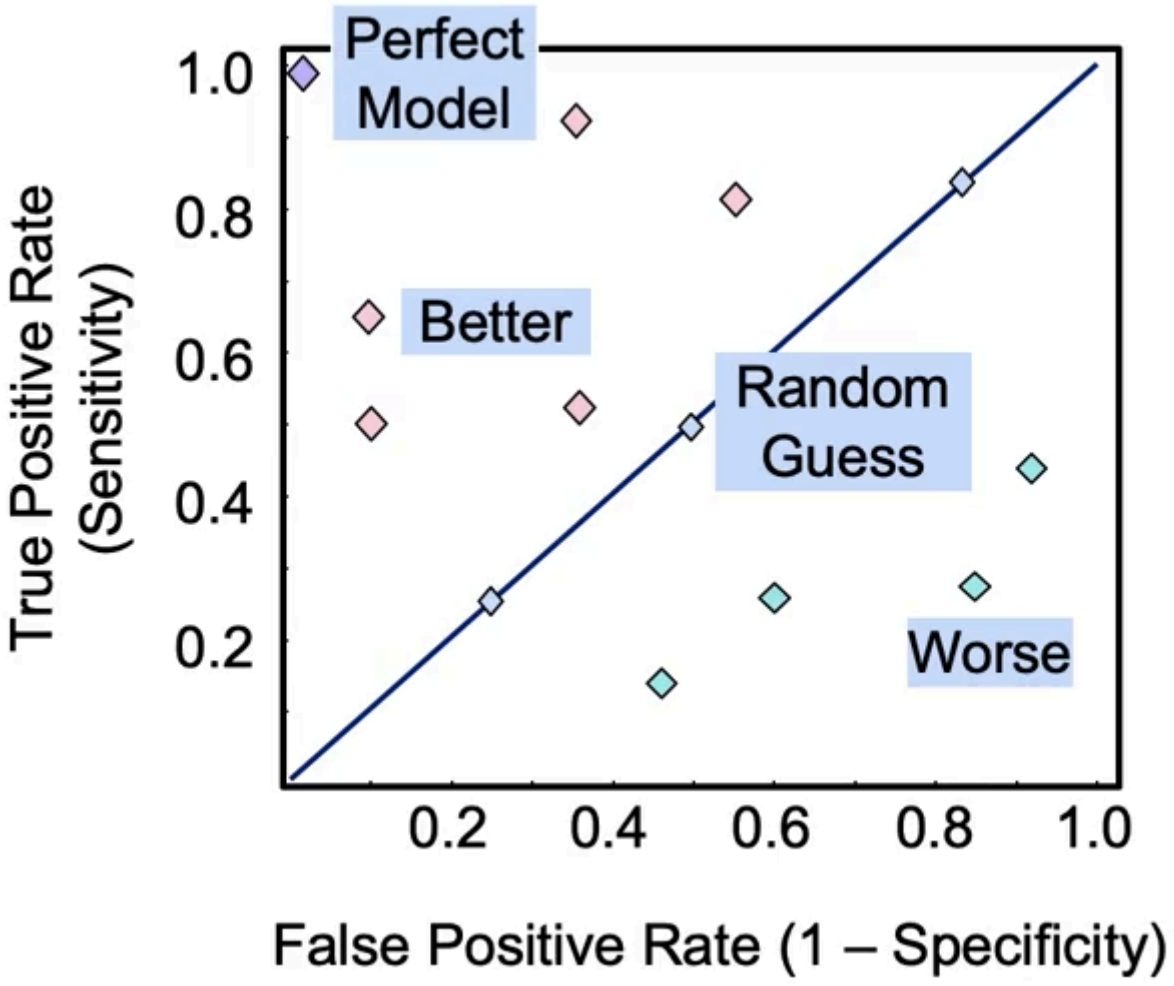
$$\textbf{Precision} = \frac{TP}{TP + FP}$$

$$\textbf{Recall or Sensitivity} = \frac{TP}{TP + FN}$$

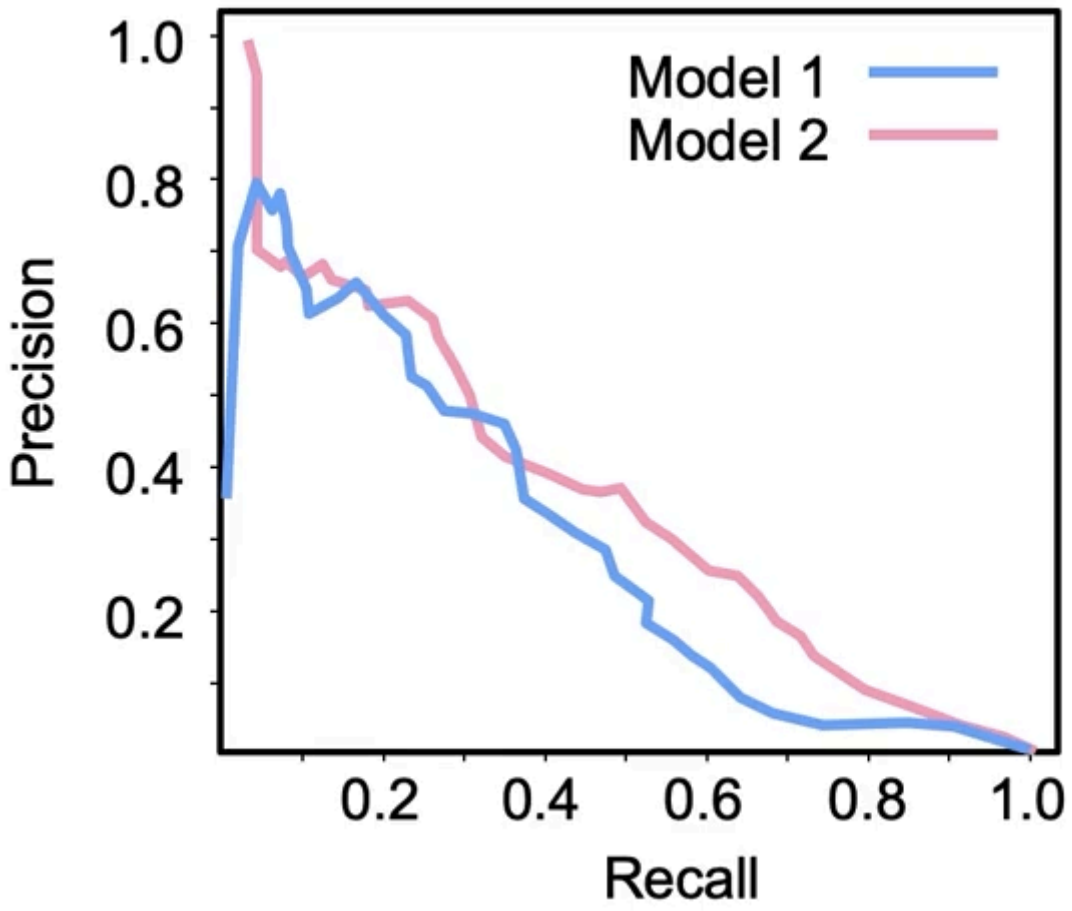
$$\textbf{Specificity} = \frac{TN}{FP + TN}$$

$$\textbf{F1} = 2\,\frac{Precision * Recall}{Precision + Recall}$$

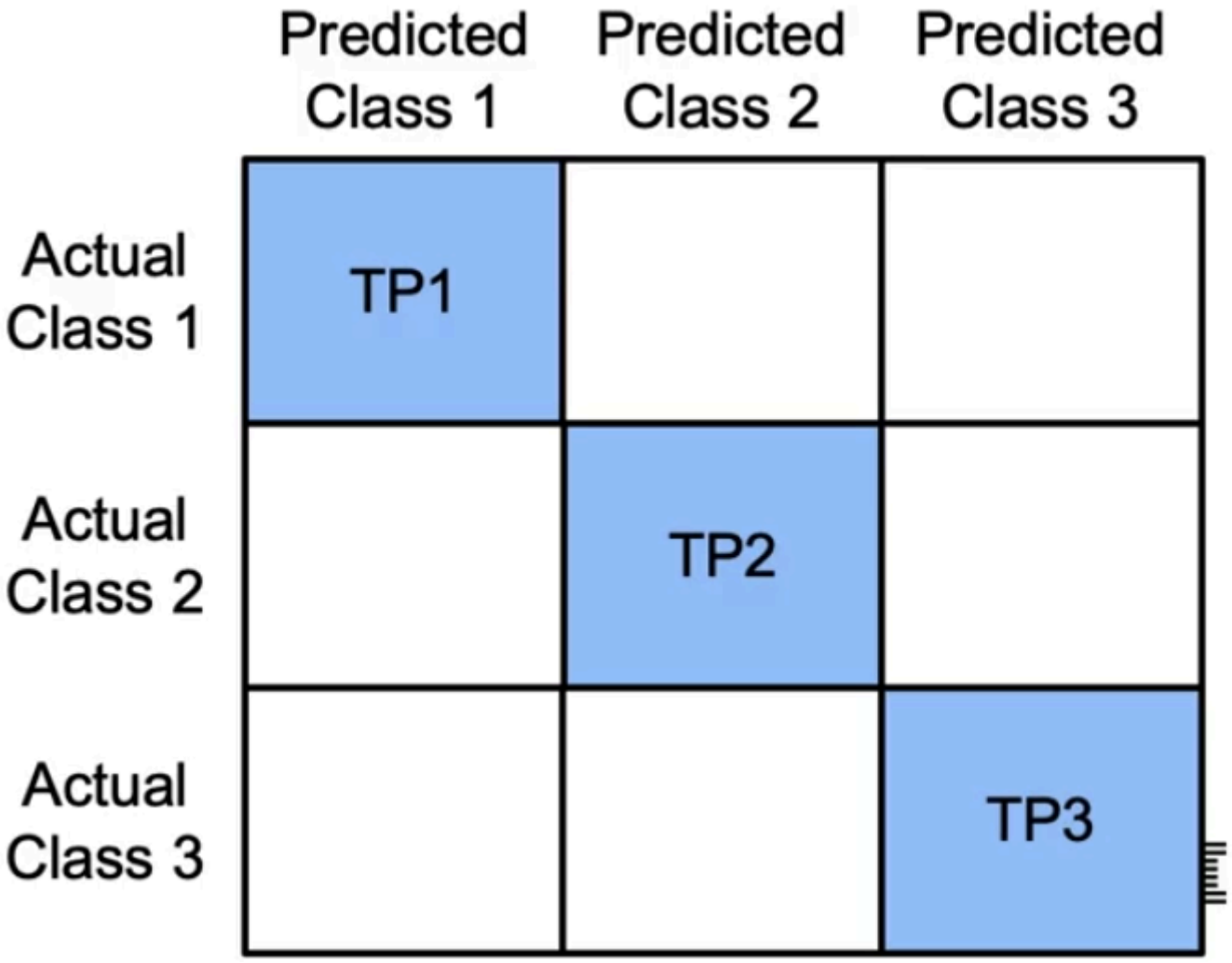
Receiver Operating Characteristic Curve (ROC Curve)



Precision-Recall Curve



Multi-class



Accuracy = (TP1+TP2+TP3) / Total

For ROC curve and Precision-recall curve, we use a one-vs-all approach

Code

```
# Lots of other error metrics and diagnostic tools:
from sklearn.metrics import precision_score, recall_score,
        f1_score, roc_auc_score,
        confusion_matrix, roc_curve,
        precision_recall_curve
```

(also `accuracy_score`)