Model Evaluation and Selection

Part 1

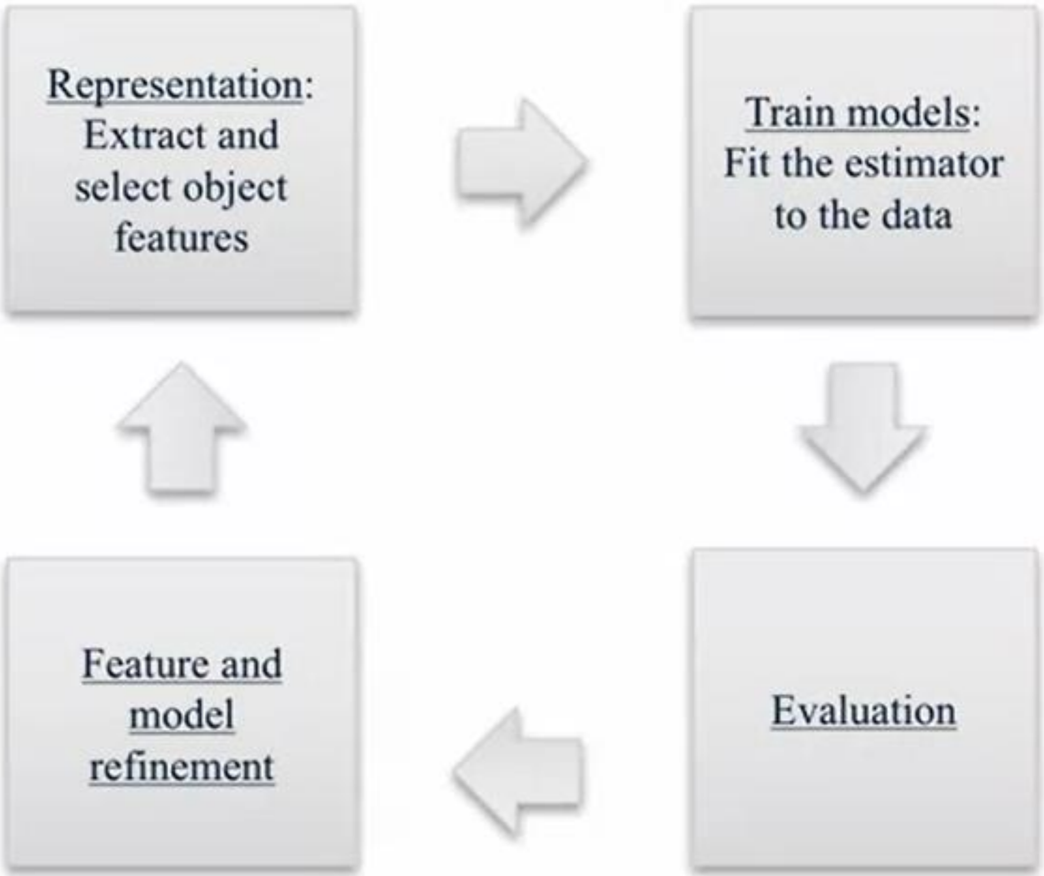
Evaluation Metrics for Classification

Learning Objectives

- Learn how to use a variety of evaluation metrics to evaluate supervised machine learning models.

- Learn about choosing the right metric for selecting between models or for doing parameter tuning.

Represent / Train / Evaluate / Refine Cycle



Evaluation

- It's very important to choose evaluation methods that match the goal of your application.
- Compute your selected evaluation metric for multiple different models.
- Then select the model with 'best' value of evaluation metric.

Accuracy with Imbalanced Classes

Like a very rare cancer

- Suppose you have two classes:
  - *Relevant (R): the positive class*
  - *Not_Relevant (N): the negative class*
- Out of 1000 randomly selected items, on average
  - *One item is relevant and has an R label*
  - *The rest of the items (999 of them) are not relevant and labelled N.*
- Recall that:

$$\text{Accuracy} = \frac{\#\text{correct predictions}}{\#\text{total instances}}$$

Accuracy with Imbalanced Classes

- You build a classifier to predict relevant items, and see that its accuracy on a test set is 99.9%.
- Wow! Amazingly good, right?
- For comparison, suppose we had a "dummy" classifier that didn't look at the features at all, and always just blindly predicted the most frequent class (i.e. the negative N class).

Accuracy with Imbalanced Classes

- Assuming a test set of 1000 instances, what would this dummy classifier's accuracy be?

- Answer:

$$\text{Accuracy}_{\text{DUMMY}} = 999 / 1000 = 99.9\%$$

Dummy classifiers completely ignore the input data!

- Dummy classifiers serve as a sanity check on your classifier's performance.
- They provide a null metric (e.g. null accuracy) baseline.
- Dummy classifiers should not be used for real problems

Dummy classifiers completely ignore the input data!

- Some commonly-used settings for the `strategy` parameter for DummyClassifier in scikit-learn:
  - *most_frequent : predicts the most frequent label in the training set.*
  - *stratified : random predictions based on training set class distribution.*
  - *uniform : generates predictions uniformly at random.*
  - *constant : always predicts a constant label provided by the user.*
    - *A major motivation of this method is F1-scoring, when the positive class is in the minority.*

What if my classifier accuracy is close to the null accuracy baseline?

This could be a sign of:

- Ineffective, erroneous or missing features
- Poor choice of kernel or hyperparameter
- Large class imbalance

Dummy Regressors

`strategy` **parameter options:**

- *mean* : predicts the mean of the training targets.
- *median* : predicts the median of the training targets.
- *quantile* : predicts a user-provided quantile of the training targets.
- *constant* : predicts a constant user-provided value.

Some examples

- Credit card transactions
- Web search
- Cancer prediction

Confusion Matricies

Binary Prediction Outcomes

	Predicted negative	**Predicted** positive
True negative	TN	FP
True positive	FN	TP

Label 1 = positive class (class of interest)

Label 0 = negative class (everything else)

TP = true positive

FP = false positive (Type I error)

TN = true negative

FN = false negative (Type II error)

Confusion Matrix for Binary Prediction Task

	Predicted negative	Predicted positive
True negative	TN = 356	FP = 51
True positive	FN = 38	TP = 5

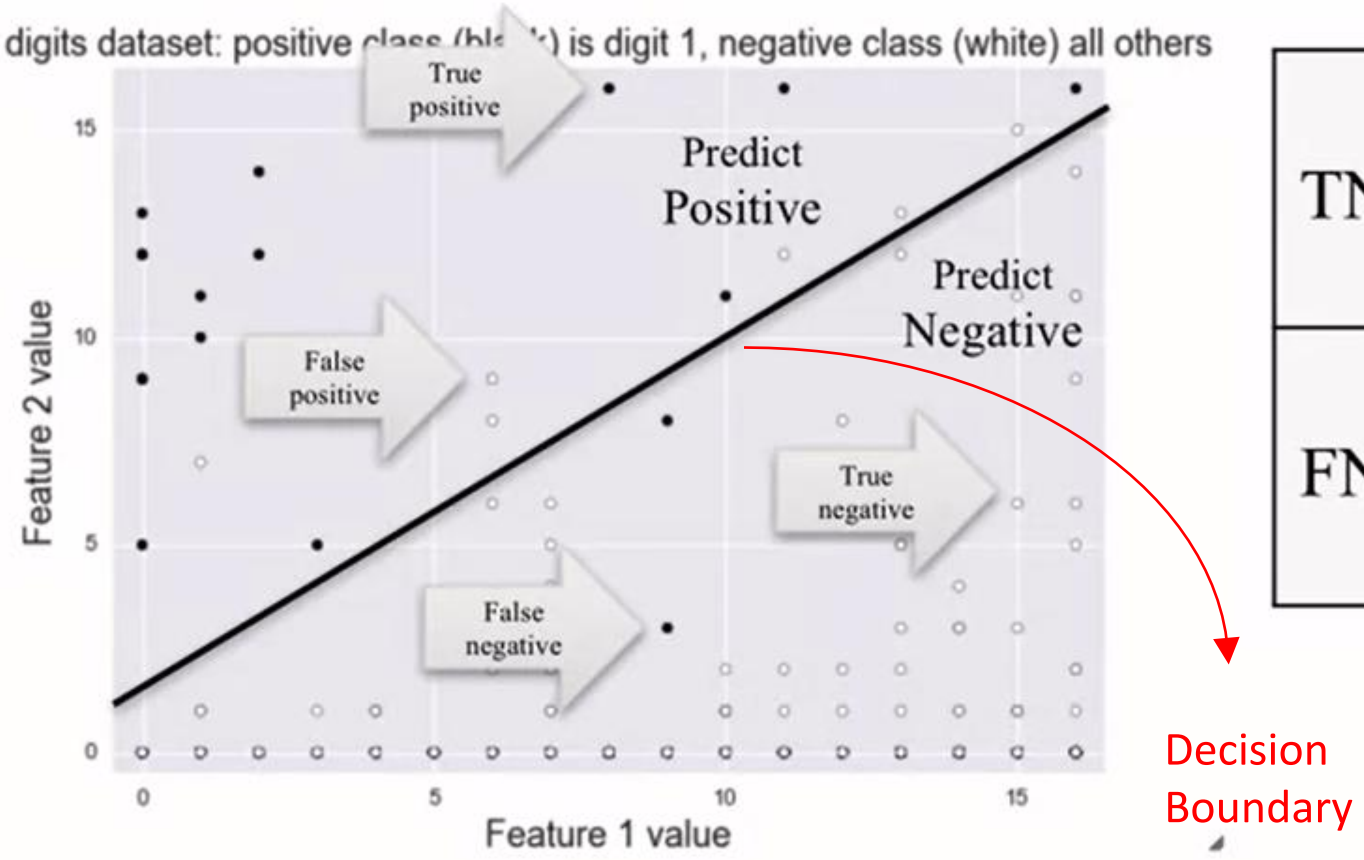
N = 450

Confusion matrix for binary prediction task

	Predicted negative	Predicted positive	
True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
			$N = 450$

Always look at the confusion matrix for your classifier.

Visualization of Different Error Types



TN = 429	FP = 6
FN = 2	TP = 13

Accuracy: for what fraction of all instances is the classifier's prediction correct (for either positive or negative class)?

True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
	Predicted negative	Predicted positive	$N = 450$

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

$$= \frac{400+26}{400+26+17+7}$$

$$= 0.95$$

Classification error (1 – Accuracy): for what fraction of all instances is the classifier's prediction incorrect?

True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
	Predicted negative	Predicted positive	$N = 450$

$$\text{ClassificationError} = \frac{FP + FN}{TN + TP + FN + FP}$$

$$= \frac{7+17}{400+26+17+7}$$

$$= 0.060$$

Recall, or True Positive Rate (TPR): what fraction of all positive instances does the classifier <u>correctly</u> identify as positive?

	Predicted negative	Predicted positive	
True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
			$N = 450$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$= \frac{26}{26+17}$$

$$= 0.60$$

Recall is also known as:

- True Positive Rate (TPR)
- Sensitivity
- Probability of detection

Precision: what fraction of positive predictions are correct?

True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
	Predicted negative	Predicted positive	$N = 450$

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

$$= \frac{26}{26+7}$$

$$= 0.79$$

Query suggestions…

False positive rate (FPR): what fraction of all negative instances does the classifier incorrectly identify as positive?

	Predicted negative	Predicted positive	
True negative	TN = 400	FP = 7	
True positive	FN = 17	TP = 26	
			$N = 450$

$$FPR = \frac{FP}{TN+FP}$$

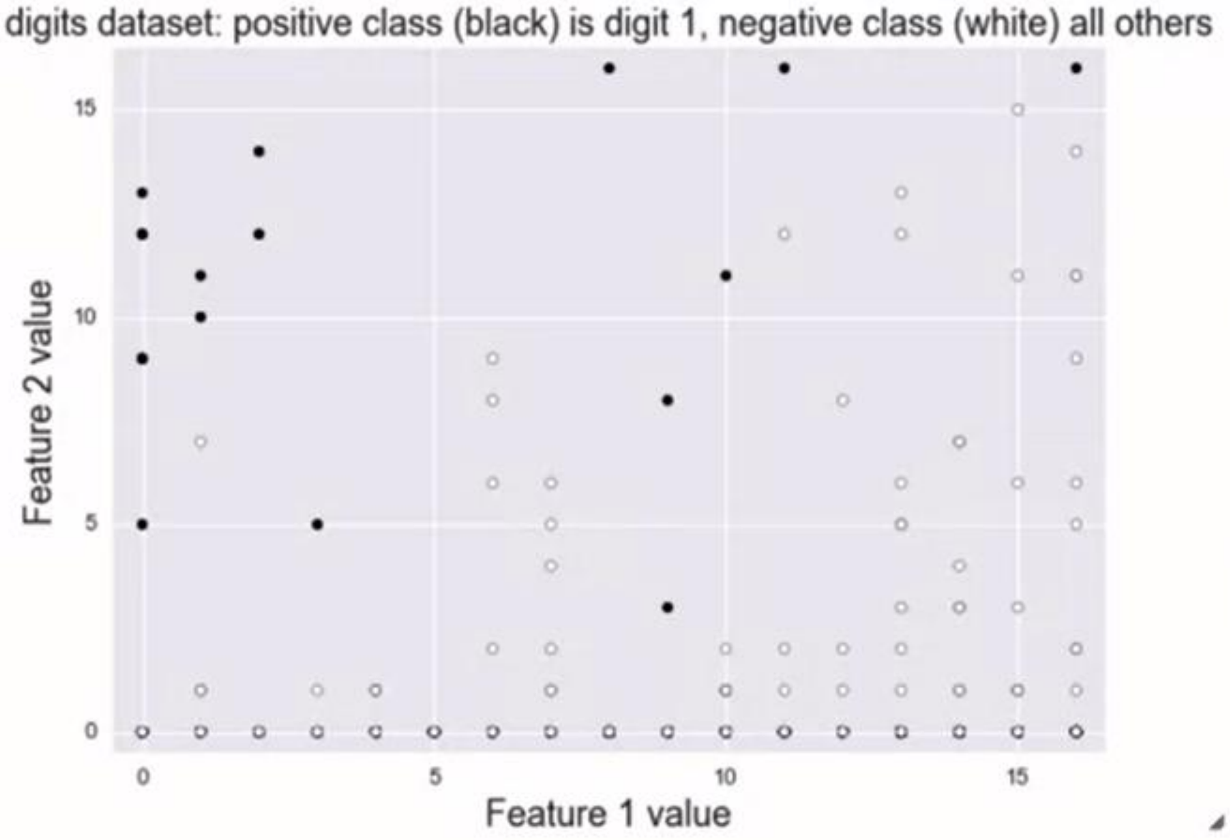
$$= \frac{7}{400+7}$$

$$= 0.02$$

False Positive Rate is also known as:

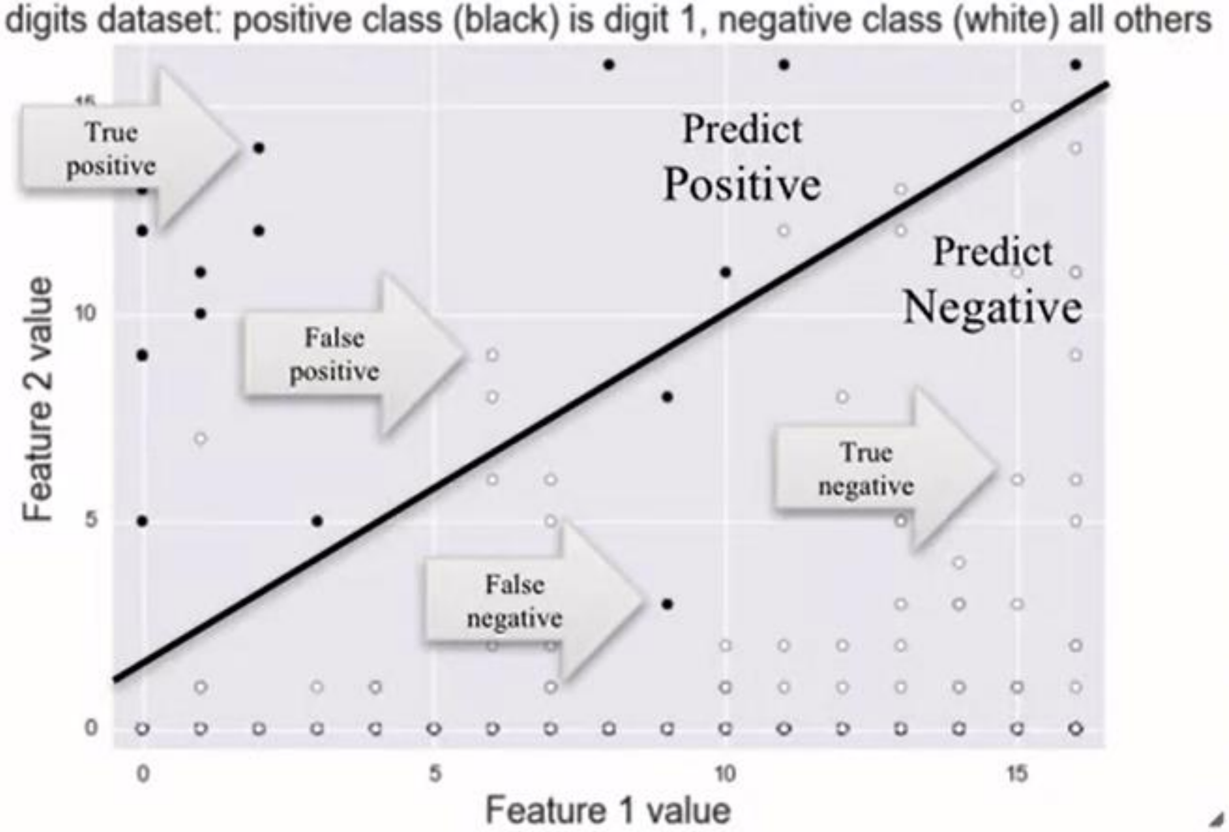
- Specificity

A Graphical Illustration of Precision & Recall



TN =	FP =
FN =	TP =

The Precision-Recall Tradeoff

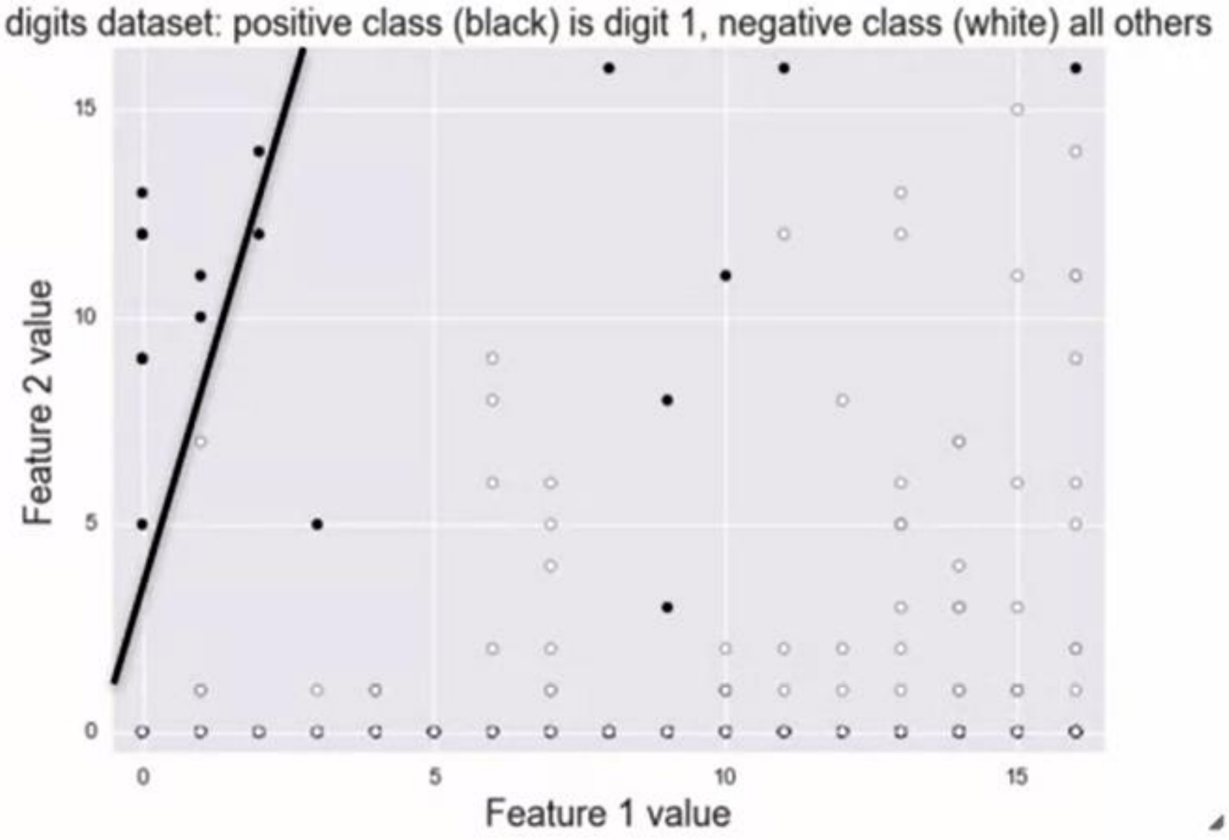


TN = 429	FP = 6
FN = 2	TP = 13

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{13}{19} = 0.68$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{13}{15} = 0.87$$

High Precision, Lower Recall



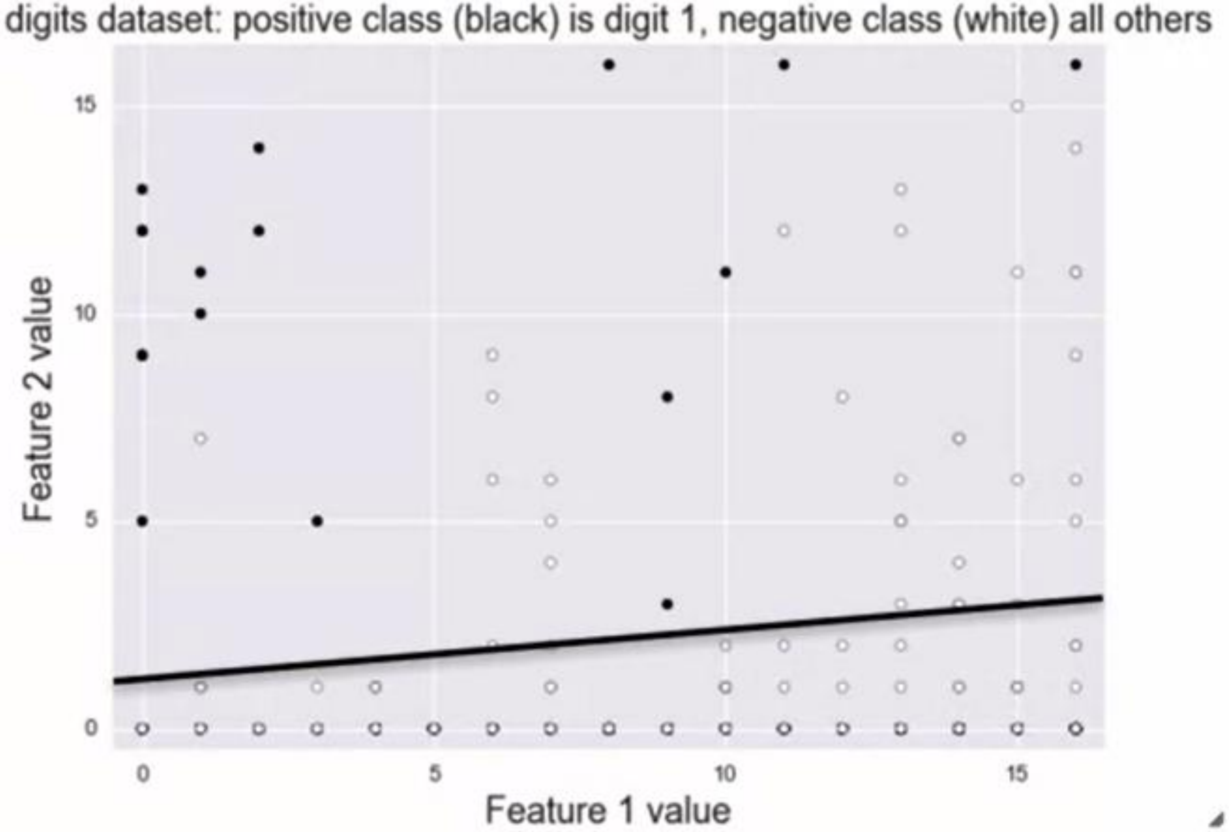
TN = 435	FP = 0
FN = 8	TP = 7

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{7}{7} = 1.00$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{7}{15} = 0.47$$

Query results.

Low Precision, High Recall



TN = 408	FP = 27
FN = 0	TP = 15

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{15}{42} = 0.36$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{15}{15} = 1.00$$

Tumor
Prediction

There is often a tradeoff between precision and recall

- **Recall-oriented machine learning tasks:**
  - Search and information extraction in legal discovery
  - Tumor detection
  - Often paired with a human expert to filter out false positives
- **Precision-oriented machine learning tasks:**
  - Search engine ranking, query suggestion
  - Document classification
  - Many customer-facing tasks (users remember failures!)

F1-score: combining precision & recall into a single number

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}$$

F-score: generalizes F1-score for combining precision & recall into a single number

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall} = \frac{(1 + \beta^2) \cdot TP}{(1 + \beta^2) \cdot TP + \beta \cdot FN + FP}$$

β allows adjustment of the metric to control the emphasis on recall vs precision:

- **Precision-oriented users: $\beta = 0.5$ (false positives hurt performance more than false negatives)**
- **Recall-oriented users: $\beta = 2$ (false negatives hurt performance more than false positives)**