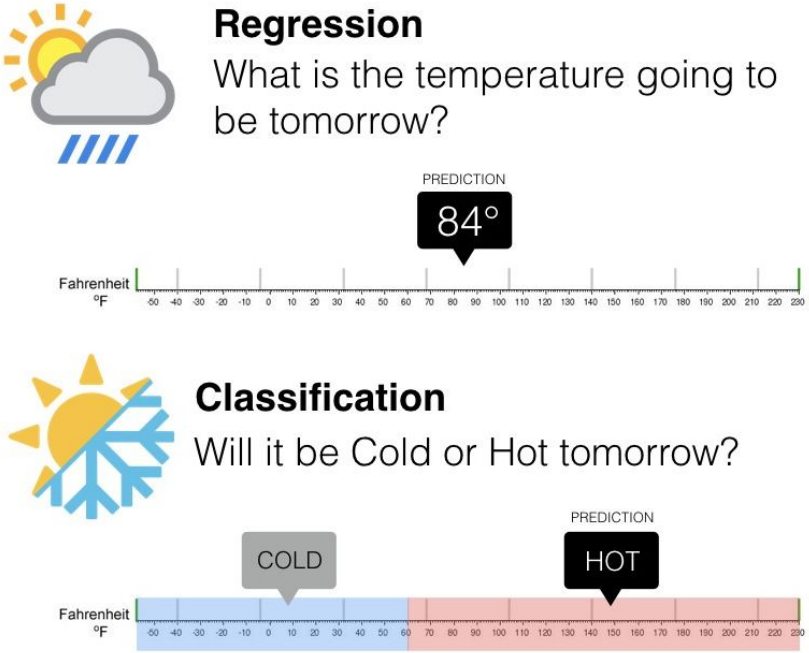
CS410: Principles and Techniques of Data Science

Module 10: Classification

https://drive.google.com/drive/folders/1SyUy583HGisF2qdaDSVsvNPT7_GyS2Oz?usp=sharing

Introduction

The most common supervised learning tasks are regression (predicting values) and classification (predicting classes).



MNIST

Digits from the MNIST dataset

There are 70,000 images in MNIST dataset.

Each image has 784 features(28 × 28 pixels)

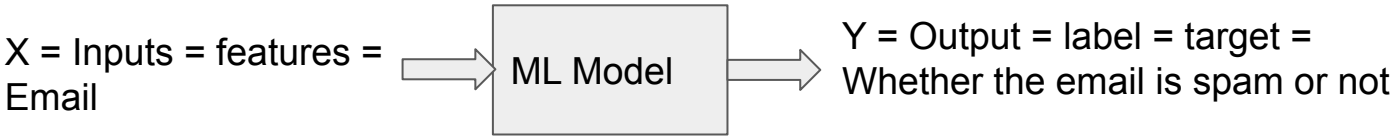
Each feature simply represents one pixel’s intensity, from 0 (white) to 255 (black).

Training a Binary Classifier: GD Classifier

Gradient Descent (GD) classifier

- Capable of handling very large datasets efficiently
-

Spam



There are 100 emails (90 ham + 10 spam): ML model v1 predicted 80 ham + 20 spam

Accuracy = correct predictions / total dataset size = (80 + 10) / 100 = 0.9

ML model 2 (dumb classifier) - has predicted everything as ham

Accuracy = 90 / 100 = 0.9

Performance Measures

Measuring Accuracy:

- Dumb classifier that just classifies every single image in the “not-5” class has over 90% accuracy!
- This is simply because only about 10% of the images are 5s, so if you always guess that an image is not a 5, you will be right about 90% of the time.

Demonstrates why **accuracy is generally not the preferred performance measure for classifiers, especially when you are dealing with skewed datasets (i.e., when some classes are much more frequent than others).**

Performance Measures: Confusion Matrix

Accuracy = correct predictions / total dataset size

= (53892 + 3530) / (53892 + 3530 + 1891 +687)

		Predicted	
		Negative (**N**) **HAM**	Positive (**P**) **SPAM**
Actual	Negative - **HAM**	True Negatives (**TN**) **53892**	False Positives (**FP**) **Type I error** **687**
	Positive + **SPAM**	False Negatives (**FN**) **Type II error** **1891**	True Positives (**TP**) **3530**

```
array([[53892,   687],
       [ 1891,  3530]])
```

Performance Measures: Confusion Matrix

```
array([[53892,   687],
       [ 1891,  3530]])
```

- Each row in a confusion matrix represents an actual class, while each column represents a predicted class.
- The first row of this matrix considers non-5 images (the negative class):
  - 53,892 of them were correctly classified as non-5s (true negatives)
  - Remaining 687 were wrongly classified as 5s (false positives).
- The second row considers the images of 5s (the positive class):
  - 1891 were wrongly classified as non-5s (false negatives)
  - Remaining 3530 were correctly classified as 5s (true positives).

Performance Measures: Confusion Matrix

A perfect classifier would have only true positives and true negatives, so its confusion matrix would have nonzero values only on its main diagonal (top left to bottom right).

```
array([[54579,     0],
       [    0,  5421]])
```

Precision

Precision = accuracy of positive predictions

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

TP is the number of true positives, and FP is the number of false positives.

Perfect precision is obtained if the classifier makes one single positive prediction and it is correct (precision = 1/1 = 100%).

Recall

Recall (sensitivity) = true positive rate
= ratio of positive instances that are correctly detected by classifier

$$\text{precision} = \frac{TP}{TP + FP} \qquad \text{recall} = \frac{TP}{TP + FN}$$

TP is the number of true positives, and FN is the number of false negatives.

If you care about FP -> precision; if you care about FN -> recall
Spam classification - Do you care about FP (falsely predicting as spam) or FN (falsely predicting as ham)? - we care about FP - precision

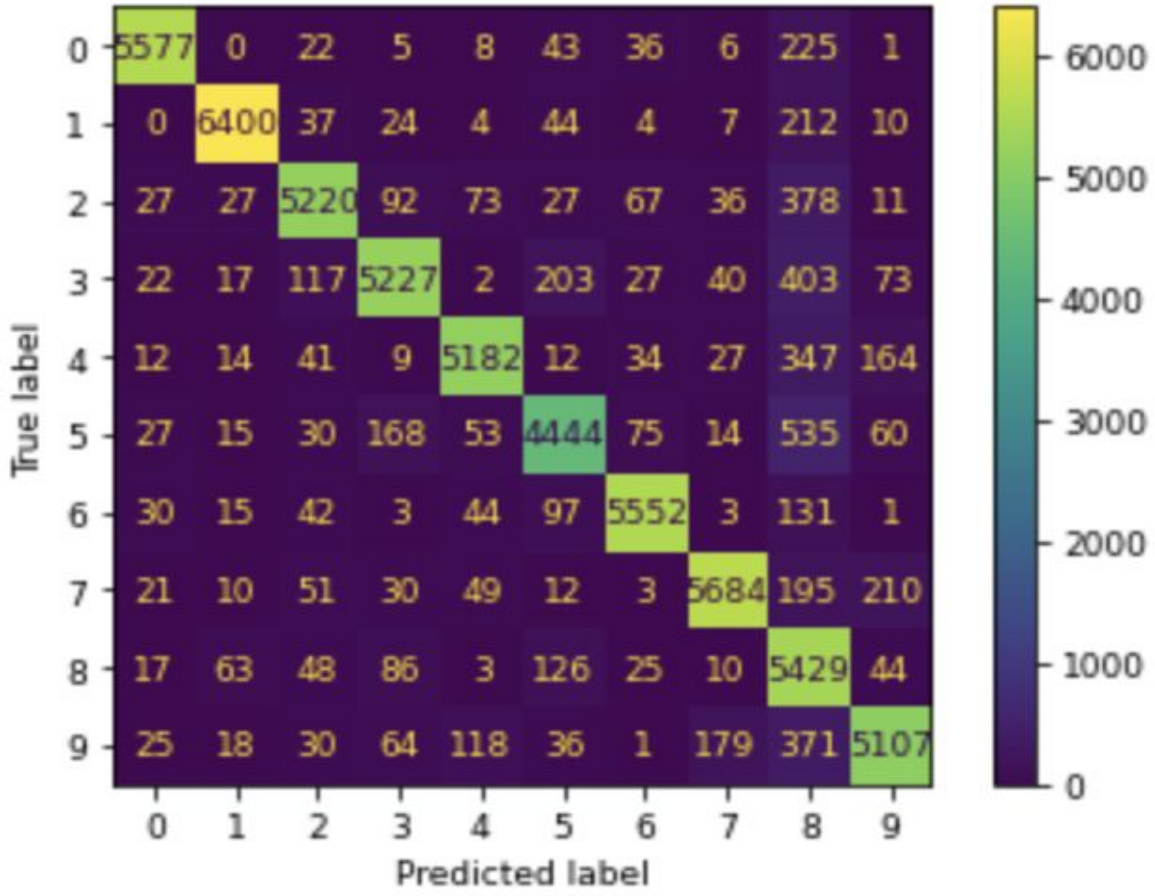
F1 score

F1 score combine precision and recall into a single metric.
It is the harmonic mean of precision and recall.

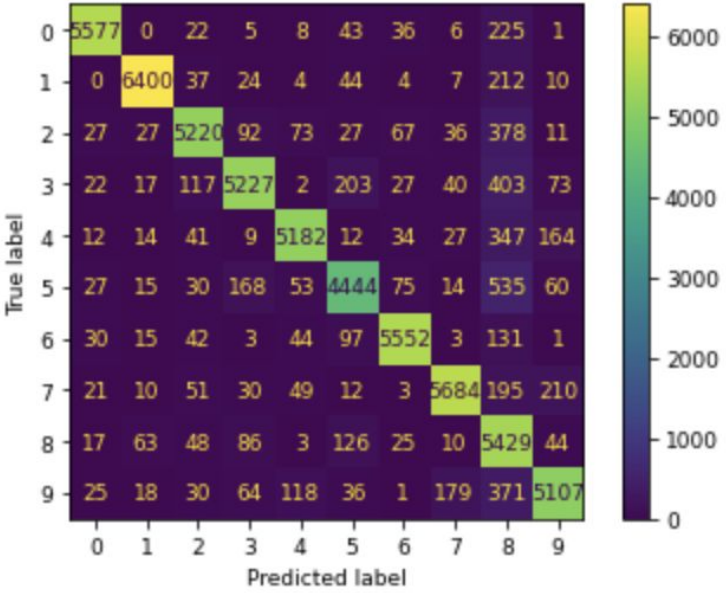
Regular mean treats all values equally, but the harmonic mean gives much more weight to low values.
Classifier will only get a high F1 score if both recall and precision are high.

$$F_1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Error Analysis



Error Analysis



Confusion matrix looks pretty good, since most images are on the main diagonal, which means that they were classified correctly.

Let's focus the plot on the errors.

You can clearly see the kinds of errors the classifier makes.
The values in the column for class 8 is quite large, which tells you that many images get misclassified as 8s. However, the row for class 8 is not that bad, telling you that actual 8s in general get properly classified as 8s.

You can also see that 3s and 5s often get confused (in both directions).

Analyzing the confusion matrix often gives you insights into ways to improve your classifier.

Looking at this plot, it seems that your efforts should be spent on reducing the false 8s. For example, you could try to gather more training data for digits that look like 8s (but are not) so that the classifier can learn to distinguish them from real 8s.

THANK YOU