CS 453X: Class 1

Jacob Whitehill

How old are these people?

Guess how old each person is — *at the time the photo was taken* — based on their face image.







https://www.vision.ee.ethz.ch/en/publications/papers/articles/eth_biwi_01299.pdf

Whose guesses were the best?

- Who of you was best at guessing the people's ages?

- How do we define "best"?

- We need some kind of **accuracy function**.

How to measure accuracy?

- Suppose there are *n* faces.
- Let $\mathbf{y} \in \mathbb{R}^n$ be an *n*-vector of the **ground-truth** age values.
- Let $\hat{\mathbf{y}} \in \mathbb{R}^n$ be an *n*-vector of **guesses** for the age values.
- We write y_i and $\hat{y}_i$ for the *i*th ground-truth and guess (i=1, …, n), respectively.
- What kinds of functions $f(\hat{\mathbf{y}}, \mathbf{y})$ might we define to express the accuracy of the guesses w.r.t. ground-truth?

How to measure accuracy?

- Percent exactly Correct: $f_{\mathrm{PC}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[y_i = \hat{y}_i]$

  where $\mathbb{I}[\cdot]$ equals 1 if the condition is true, and 0 otherwise.

- **Higher** scores are better.

How to measure accuracy?

- Percent exactly Correct: $f_{\mathrm{PC}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[y_i = \hat{y}_i]$

  where $\mathbb{I}[\cdot]$ equals 1 if the condition is true, and 0 otherwise.

- **Example**:

  $\mathbf{y}$ = [66, 18, 38, 61]

  $\hat{\mathbf{y}}^{(1)}$ = [65, 19, 38, 60]

  $\hat{\mathbf{y}}^{(2)}$ = [5, 6, 2, 61]

  $f_{\mathrm{PC}}(\mathbf{y}, \hat{\mathbf{y}}^{(1)}) =$ **?**

  $f_{\mathrm{PC}}(\mathbf{y}, \hat{\mathbf{y}}^{(2)}) =$ **?**

How to measure accuracy?

- Percent exactly Correct: $f_{\mathrm{PC}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[y_i = \hat{y}_i]$

  where $\mathbb{I}[\cdot]$ equals 1 if the condition is true, and 0 otherwise.

- **Example**:

  $\mathbf{y}$ = [66, 18, 38, 61]

  $\hat{\mathbf{y}}^{(1)}$ = [65, 19, 38, 60]

  $\hat{\mathbf{y}}^{(2)}$ = [5, 6, 2, 61]

  $$f_{\mathrm{PC}}(\mathbf{y}, \hat{\mathbf{y}}^{(1)}) = 0.25$$

  $$f_{\mathrm{PC}}(\mathbf{y}, \hat{\mathbf{y}}^{(2)}) = 0.25$$

  Problem: It gives no consideration to the *distance* from ground-truth.

How to measure accuracy?

- Percent exactly Correct: $f_{\mathrm{PC}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[y_i = \hat{y}_i]$

  where $\mathbb{I}[\cdot]$ equals 1 if the condition is true, and 0 otherwise.

- **Example**:
  $\mathbf{y}$ = [66, 18, 38, 61]
  $\hat{\mathbf{y}}^{(1)}$ = [65, 19, 38, 60]
  $\hat{\mathbf{y}}^{(2)}$ = [5, 6, 2, 61]

- For a **regression** problem such as ours — in which we the ground-truth can be any real number — f_{PC} is almost never used.

How to measure accuracy?

- Average distance: $f_{\text{avgdist}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$
  - **Lower** scores are better.

How to measure accuracy?

- Average distance: $f_{\text{avgdist}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$

- **Example**:

  $\mathbf{y}$ = [66, 18, 38, 61]

  $\hat{\mathbf{y}}^{(1)}$ = [65, 17, 39, 64]

  $\hat{\mathbf{y}}^{(2)}$ = [5, 79, 34, 69]

  $f_{\text{avgdist}}(\mathbf{y}, \hat{\mathbf{y}}^{(1)}) =$ **?**

  $f_{\text{avgdist}}(\mathbf{y}, \hat{\mathbf{y}}^{(2)}) =$ **?**

How to measure accuracy?

- Average distance: $$f_{\text{avgdist}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$$
$$= \underbrace{\frac{1}{n}\sum_{i=1}^{n} y_i}_{\text{average ground-truth}} - \underbrace{\frac{1}{n}\sum_{i=1}^{n} \hat{y}_i}_{\text{average guess}}$$

- **Example**:

  $\mathbf{y}$ = [66, 18, 38, 61]

  $\hat{\mathbf{y}}^{(1)}$ = [65, 17, 39, 64]

  $\hat{\mathbf{y}}^{(2)}$ = [5, 79, 34, 69]

  $$f_{\text{avgdist}}(\mathbf{y}, \hat{\mathbf{y}}^{(1)}) = \frac{1}{4}(1 + 1 - 1 - 3) = -0.5 \text{ yr}$$

  $$f_{\text{avgdist}}(\mathbf{y}, \hat{\mathbf{y}}^{(2)}) = \frac{1}{4}(60 - 62 + 5 - 5) = -0.5 \text{ yr}$$

  Problem: It only measures whether the *average guess* is close to the *average ground-truth* (i.e., *bias*). Individual distances are averaged out.

How to measure accuracy?

- Mean absolute error: $f_{\mathrm{MAE}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
  - **Lower** scores are better.
  - **Example**:

    $\mathbf{y}$ = [66, 18, 38, 61]

    $\hat{\mathbf{y}}^{(1)}$ = [65, 17, 39, 64]

    $\hat{\mathbf{y}}^{(2)}$ = [5, 79, 34, 69]

    $f_{\mathrm{MAE}}(\mathbf{y}, \hat{\mathbf{y}}^{(1)}) =$ **?**

    $f_{\mathrm{MAE}}(\mathbf{y}, \hat{\mathbf{y}}^{(2)}) =$ **?**

How to measure accuracy?

- Mean absolute error: $f_{\mathrm{MAE}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
  - **Lower** scores are better.
  - **Example**:

    $\mathbf{y}$ = [66, 18, 38, 61]

    $\hat{\mathbf{y}}^{(1)}$ = [65, 17, 39, 64]

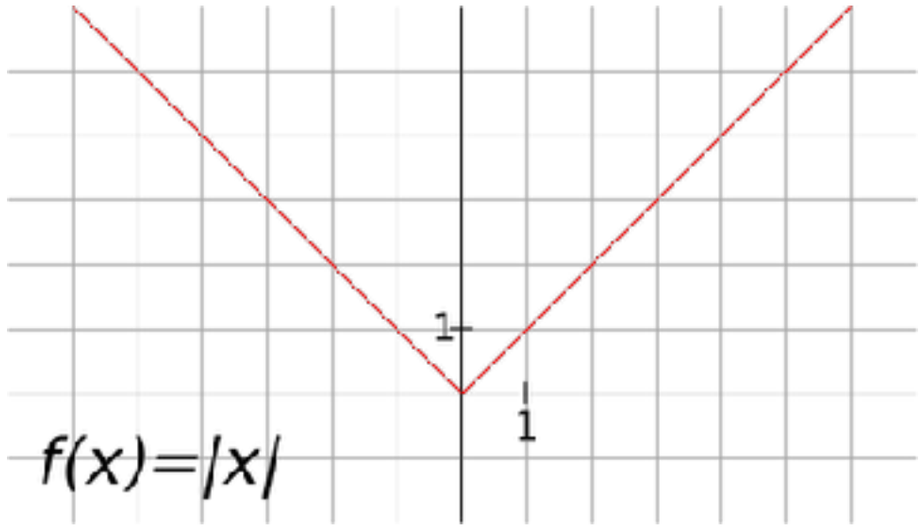
    $\hat{\mathbf{y}}^{(2)}$ = [5, 79, 34, 69]

    $$f_{\mathrm{MAE}}(\mathbf{y}, \hat{\mathbf{y}}^{(1)}) = \frac{1}{4}(1+1+1+3) = 1.5 \text{ yr}$$

    $$f_{\mathrm{MAE}}(\mathbf{y}, \hat{\mathbf{y}}^{(2)}) = \frac{1}{4}(60+62+5+5) = 23 \text{ yr}$$

How to measure accuracy?

- Mean absolute error: $f_{\mathrm{MAE}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$
  - **Issue**: Absolute value is not differentiable at 0. This is important since much of ML involves differential calculus.



https://commons.wikimedia.org/wiki/File:F(x)%3DAbs(x).svg

How to measure accuracy?

- Mean squared error: $f_{\mathrm{MSE}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
  - **Lower** scores are better.

$\mathbf{y}$ = [66, 18, 38, 61]

$\hat{\mathbf{y}}^{(1)}$ = [65, 17, 39, 64]

$\hat{\mathbf{y}}^{(2)}$ = [5, 79, 34, 69]

$$f_{\mathrm{MSE}}(\mathbf{y}, \hat{\mathbf{y}}^{(1)}) = \frac{1}{4}(1^2 + 1^2 + 1^2 + 3^2) = 3 \text{ yr}^2$$

$$f_{\mathrm{MSE}}(\mathbf{y}, \hat{\mathbf{y}}^{(2)}) = \frac{1}{4}(60^2 + 62^2 + 5^2 + 5^2) = 1873.5 \text{ yr}^2$$

How to measure accuracy?

- Mean squared error: $f_{\mathrm{MSE}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
  - f_{MSE} is also differentiable.
  - **Note**: f_{MSE} expresses the error in *squared* units.

How to measure accuracy?

- Root mean squared error: $f_{\mathrm{RMSE}}(\mathbf{y}, \hat{\mathbf{y}}) = \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \right)^{1/2}$
  - **Lower** scores are better.

$\mathbf{y}$ = [66, 18, 38, 61]

$\hat{\mathbf{y}}^{(1)}$ = [65, 17, 39, 64]

$\hat{\mathbf{y}}^{(2)}$ = [5, 79, 34, 69]

$$f_{\mathrm{RMSE}}(\mathbf{y}, \hat{\mathbf{y}}^{(1)}) = \left(\frac{1}{4}(1^2 + 1^2 + 1^2 + 3^2) \right)^{1/2} \approx 1.73 \text{ yr}$$

$$f_{\mathrm{RMSE}}(\mathbf{y}, \hat{\mathbf{y}}^{(2)}) = \left(\frac{1}{4}(60^2 + 62^2 + 5^2 + 5^2) \right)^{1/2} \approx 43.28 \text{ yr}$$

How to measure accuracy?

- Root mean squared error: $f_{\mathrm{RMSE}}(\mathbf{y}, \hat{\mathbf{y}}) = \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \right)^{1/2}$

  Note: f_{RMSE} function expresses the error in the same *scale* as the ground-truth: years (not years^2). However, in practice, f_{MSE} is more commonly used:

  - Square root is tedious.
  - Monotonic relationship:

    $\hat{\mathbf{y}}^{(1)}$ is better than $\hat{\mathbf{y}}^{(2)}$ in terms of RMSE $\Leftrightarrow$

    $\hat{\mathbf{y}}^{(1)}$ is better than $\hat{\mathbf{y}}^{(2)}$ in terms of MSE

How to measure accuracy?

- In general, there are many ways of measuring accuracy.
- The appropriate metric depends on the kind of task (e.g., regression, classification) and the application domain.
- **Regression**:
  - Ground-truth values are usually elements of an infinite set (e.g., real numbers).
  - We care more about distance between guess and ground-truth than exact match.
  - Example: estimate age from a face image

How to measure accuracy?

- In general, there are many ways of measuring accuracy.

- The appropriate metric depends on the kind of task (e.g., regression, classification) and the application domain.

- **Classification**:

  - Ground-truth values are usually elements of a finite set (e.g., {0,1}, {-1,+1}, {😀, 🧐, 😖, 😡}).

  - Example: estimate emotion from a face image.

How to measure accuracy?

- Some people define “accuracy” as only f_{PC}.

- Many other people (including me) treat “accuracy” as a broad family of functions:

  - When the accuracy measures the error (i.e., lower is better), it is often called a **cost** or a **loss**.

  - Loss, cost (e.g., f_{MSE}, f_{MAE}): want to *minimize*.

  - Accuracy (e.g., f_{PC}, f_{AUC}): want to *maximize*.

How old are these people?

Guess how old each person is — *at the time the photo was taken* — based on their face image.

66

18

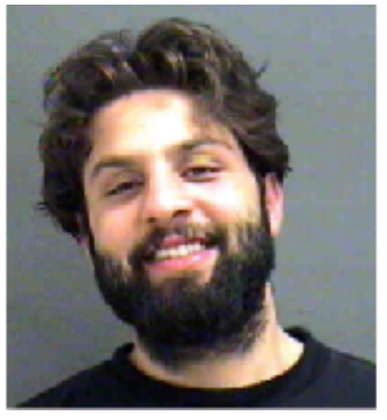
38

61

57

53

29



23

https://www.vision.ee.ethz.ch/en/publications/papers/articles/eth_biwi_01299.pdf

Age estimation accuracy

- Ground-truth **y** = [66, 18, 38, 61, 57, 53, 29, 23]
- Compute your own MSE on these 8 images:

$$f_{\mathrm{MSE}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Who had the best (lowest) score?

Age estimation accuracy

- What if we compute the **average** of the different predictions?

$$\overline{\mathbf{y}} = \frac{1}{m}\sum_{j=1}^{m}\hat{\mathbf{y}}^{(j)}$$

- What is the MSE of the **average predictor**?

$$f_{\mathrm{MSE}}(\overline{\mathbf{y}})$$

- How does this compare with the **average MSE** of all the predictors?

$$\frac{1}{m}\sum_{j=1}^{m} f_{\mathrm{MSE}}(\hat{\mathbf{y}}^{(j)})$$

Python and ML

- Python is one of the most popular languages for machine learning (ML).
- Advantages:
    - Interpreted — easy for debugging
    - High level of abstraction to accomplish large amounts of computation with concise syntax
    - Excellent library support for scientific computing & ML.
- Disadvantage:
    - Slower than C for many programming tasks.

Vectorization

- Computational bottleneck in Python: **iteration is slow**.

- To avoid large-scale iteration, we can often perform the same operation using matrix arithmetic.

    - With libraries such as **`numpy`**, Python can "offload" tedious computation to a high-performance low-level library for matrix manipulation (BLAS, CUDA, etc.).

    - This conversion is sometimes called **vectorization**.

- GPU-based fast linear algebra routines have resulted in big gains for machine learning applications.

Vectorization

- Example: dot product between two large vectors **x**, **y**

  - Iterative:

```
def dot (x, y):  # x, y are Python lists
  total = 0
  for i in range(len(x)):
    total += x[i] * y[i]
  return total
```

  - Vectorized:

```
def dot (x, y):  # x, y are numpy arrays
  return x.dot(y)  # Much faster!
```

Vectorization

- Similarly, we can vectorize the MSE computation:

$$f_{\text{MSE}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}(\mathbf{y} - \hat{\mathbf{y}})^\top(\mathbf{y} - \hat{\mathbf{y}})$$

Age estimation accuracy

- Show **`wisdom.py`**

Age estimation accuracy

- Show **`wisdom.py`**
- The MSE of the average predictor tends to be lower (better) than the average MSE over all predictors.
- This is an instance of the “wisdom of the crowd”.
- Averaging together multiple predictor is sometimes called an **ensemble**.

Who is smiling?

Which of these people are smiling?








https://www.vision.ee.ethz.ch/en/publications/papers/articles/eth_biwi_01299.pdf

Who is smiling?

Which of these people are smiling?

1

1

0

?

0

0

0

1

https://www.vision.ee.ethz.ch/en/publications/papers/articles/eth_biwi_01299.pdf

Defining ground-truth

- Does the person *look* like they're smiling?
- Does the person *her/himself* report that they're smiling?
- Is the person's lip-corner-puller *muscle* activated?

Quantifying uncertainty

- Sometimes the ground-truth value is unclear.

- To express a “soft” belief about the ground-truth, we can use **probabilities**.

- There are a couple of ways we could do this…

Quantifying uncertainty

- **Frequentist** probabilities:
    - Ask a large group of randomly selected people to **label** the face as smiling or not.
    - Count the number of labels for “smile” and divide by the total number of labels.
    - The ratio is the probability of “smile" for that face image.

Quantifying uncertainty

- **Bayesian** probabilities (“beliefs”):
  - Ask one person how much she/he believes the image is smiling, quantified as a number between 0 and 1.
  - The “belief” score is the probability of “smile" for that face image.

Quantifying uncertainty

- Many a debate has been waged about the validity of Bayesian versus frequentist probabilities.

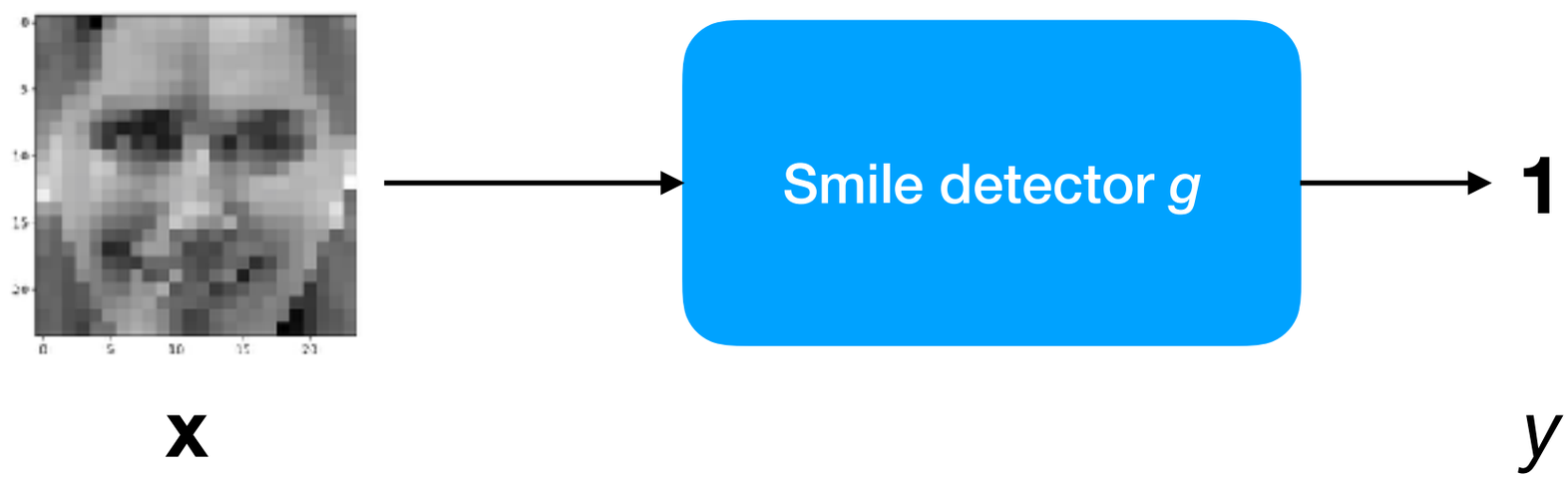
- For the most part we will steer clear of these debates and use both tools when they are useful.

Automatic smile detection

- Suppose we want to build an automatic smile detector that analyzes a grayscale face image (24x24 pixels) and reports whether the face is smiling.

- We can represent the detector as a function *g* that takes an image **x** as an input and produces a guess $\hat{y}$ as output, where $\mathbf{x} \in \mathbb{R}^{24\times 24}, \hat{y} \in \{0, 1\}$.

- Abstractly, *g* can be considered a “machine”:

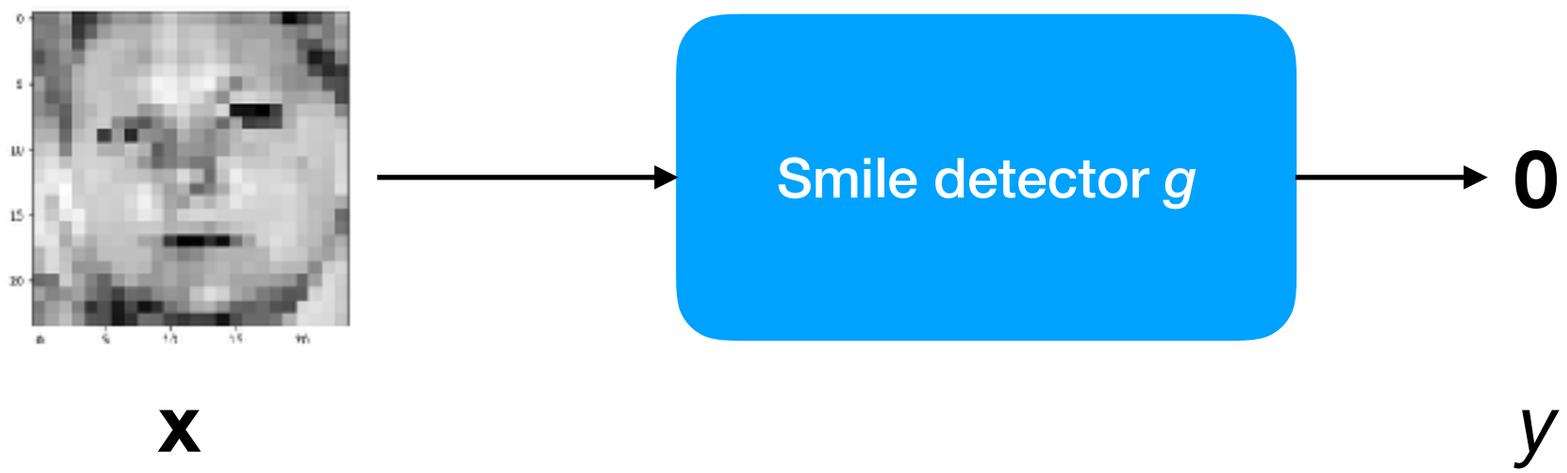


Automatic smile detection

- Suppose we want to build an automatic smile detector that analyzes a grayscale face image (24x24 pixels) and reports whether the face is smiling.

- We can represent the detector as a function *g* that takes an image **x** as an input and produces a guess $\hat{y}$ as output, where $\mathbf{x} \in \mathbb{R}^{24\times 24}, \hat{y} \in \{0, 1\}$.

- Abstractly, *g* can be considered a “machine”:



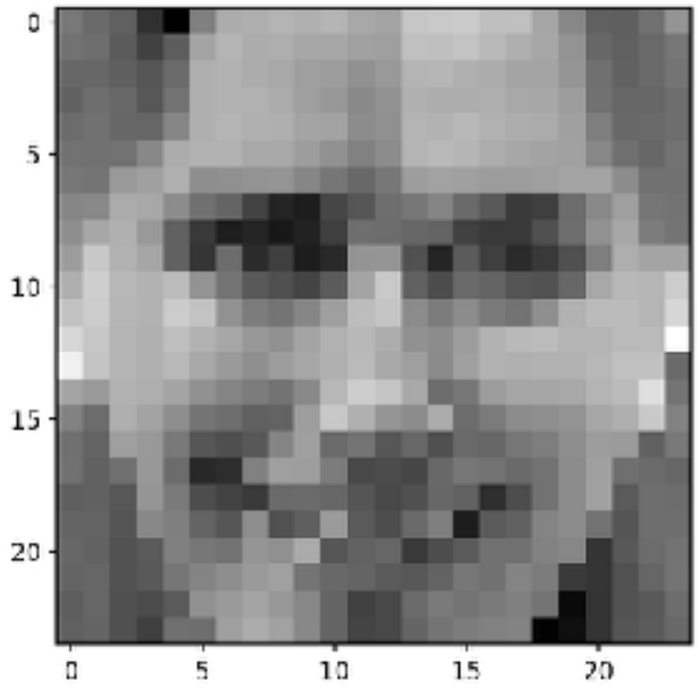
Automatic smile detection

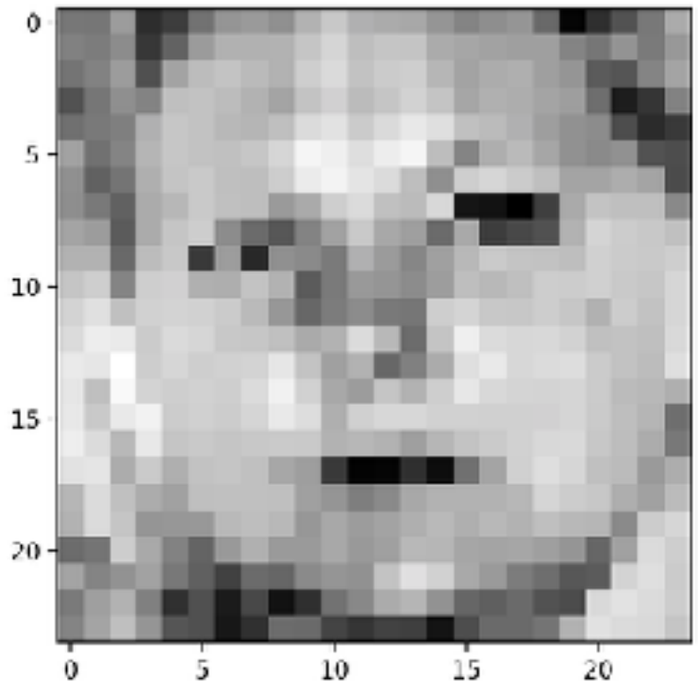
- Suppose we build *g* so that its output depends on only a *single* pair of pixels within the input face:

$$g(\mathbf{x}) = \mathbb{I}[\mathbf{x}_{r_1,c_1} > \mathbf{x}_{r_2,c_2}]$$

- Which pairs (r_1, c_1), (r_2, c_2) would you choose?

- How good is it?





Accuracy measurement

- To evaluate the simple “smile detector” *g*, we need a **test set** $\mathcal{D}^{test} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ consisting of:
  - Images { $\mathbf{x}_i$ } of both classes (smile/1, non-smile/0)
  - Corresponding labels { y_i }.
- The exact composition (ratio of positive/negative examples) is flexible but may impact the way we interpret the machine’s accuracy.

Accuracy measurement

- Let’s try a few examples by hand in smile_demo.py
- What accuracy did we achieve with a single predictor?
- Is this “good”?

Selecting a baseline

- In addition to defining an accuracy function, it's important to choose a "baseline" to which to compare your machine.

- The baseline is often the "leading brand" — the best machine that anyone has ever created before for the same problem.

- For a *new* ML problem, we might just compare to (1) random guessing or (2) selecting the most probable class based on the statistics of the dataset.

Selecting a baseline

- What fraction of faces in $\mathcal{D}^{\text{test}}$ are smiling faces? 54.6%
- How accurate (f_{PC}) would a predictor be that just always output 1 no matter what the image looked like?

Selecting a baseline

- What fraction of faces in $\mathcal{D}^{\text{test}}$ are smiling faces? 54.6%
- How accurate (f_{PC}) would a predictor be that just always output 1 no matter what the image looked like?
  - 54.6%
- Note that there are other accuracy functions (e.g., f_{AUC}) that are invariant to the proportion of each class — aka the **prior probabilities** — of each class in the test set.