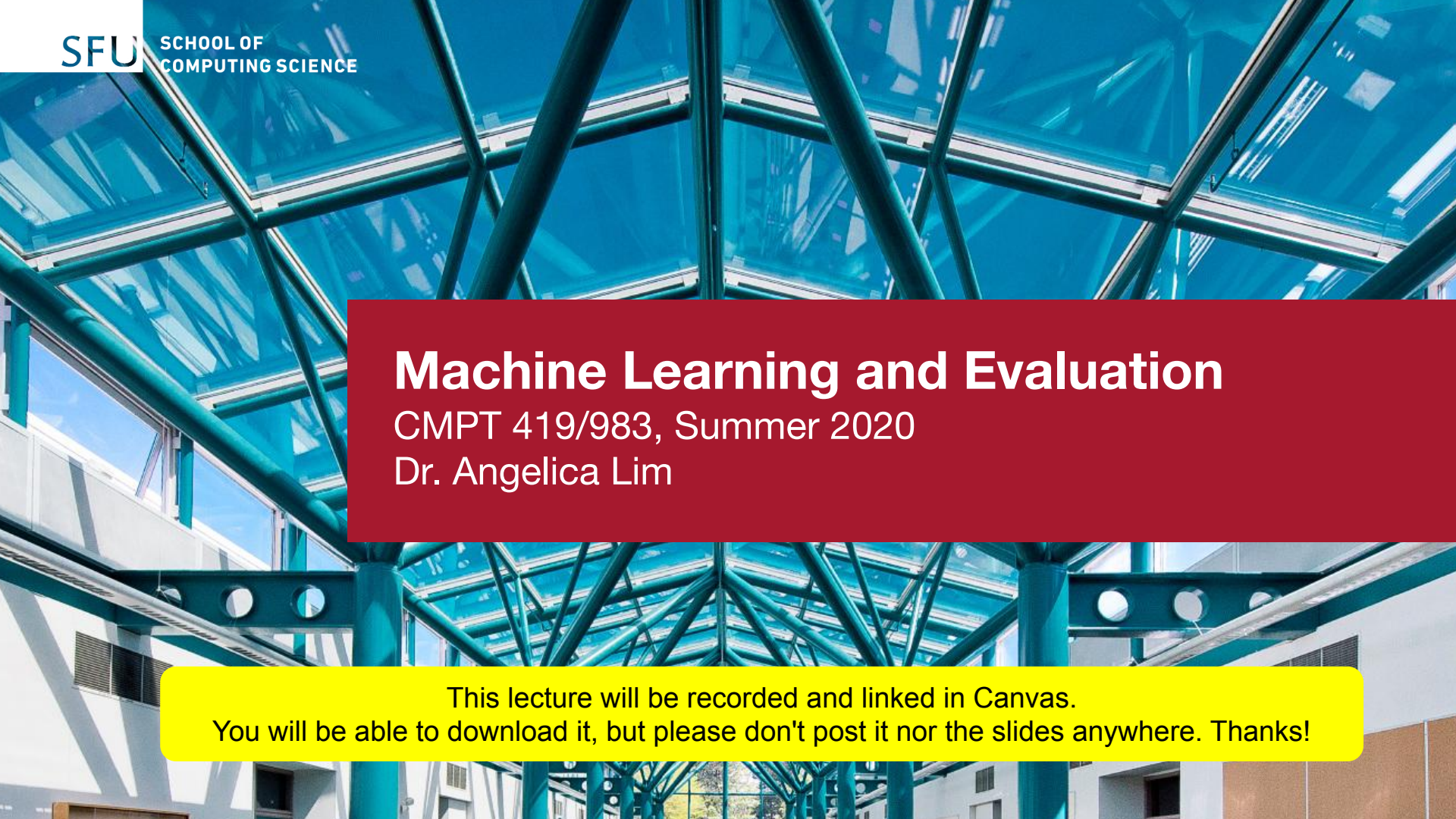


Machine Learning and Evaluation

CMPT 419/983, Summer 2020

Dr. Angelica Lim

This lecture will be recorded and linked in Canvas.
You will be able to download it, but please don't post it nor the slides anywhere. Thanks!

Activity Debrief

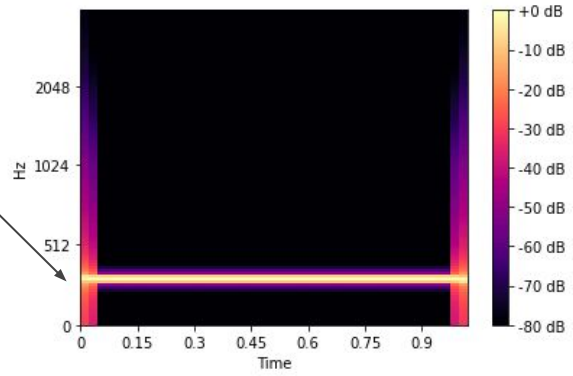
Example: Target Frequency at 440Hz

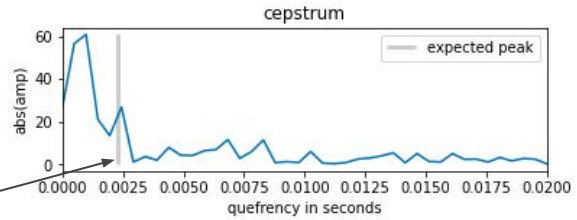
What is **quefrency**?
The inverse of frequency, measured in seconds.

What does the **cepstrum** show?
Peaks at specific quefrencies that represent the distance between peaks in the frequency spectrum.

1/440 = 0.0022

Cepstrum plot is produced with *complex_cepstrum* function in python-acoustics libary.





1. The spectogram shows high amplitude at frequency around 440Hz and not much everywhere else, which matches our target fundamental frequency set in Step 1.

2. The cepstrum plot has a peak close to quefrency ~= 0s as briefly expained in this blog post, citing the 1984 paper by Noll. Another peak is found around quefrency 1/440 ~= 0.0022s, as expected. Pretty cool!

http://flothesof.github.io/cepstrum-pitch-tracking.html

Activity Debrief

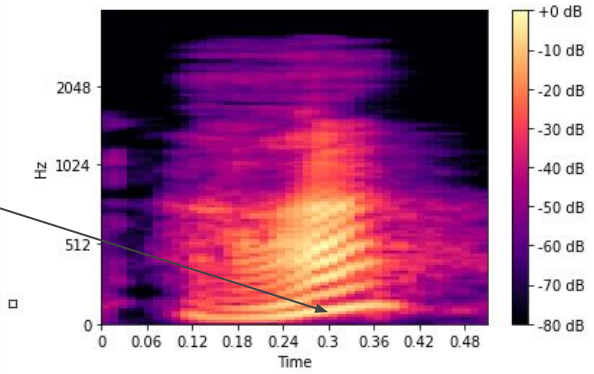
Example: Formants ~100Hz and ~250Hz

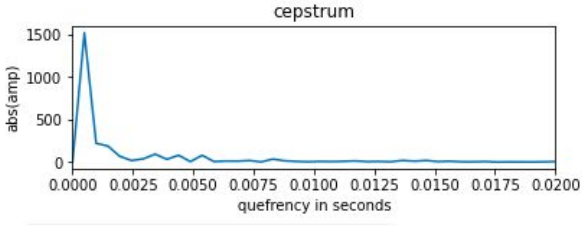
Why use the **cepstrum** when we have the **spectrogram**?

- While we can visually inspect a spectrogram, it's not very handy for getting the formants from the image
- The cepstrum allows us to do "peak-picking" (thresholding) and we can get the formants directly

A couple possible explanations for this result:

- Was the cepstrum calculated on a short time window at 0.3s?
- Filterbanks are sometimes used to "enhance" peaks





1. Sound: "hah?" with a questioning tone, male voice.

2. Formants: At time 0.3s, F1 is around 100Hz, F2 around 250Hz, making a difference of 150. Based on the same chart above, the closest vowel we have is the low, back, open [ɑ].

3. Cepstrum plot: Similarly, I expected peaks at around 1/100 = 0.01s and 1/250 = 0.004s. There are a couple of small peaks on the plot, one does look to be around 0.004s quefrency but none is at 0.01s.

https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

This Week

Tuesday

- Online activity debrief
- Assignment 3 sneak preview
- Machine Learning review
- F1 scores, accuracy, precision
- Cross-validation and leave-one-out
- Project overview

Assignment 3

A3 Preview

Description

In this assignment, you will analyze dynamic image data, i.e. video clips, using Dynamic Time Warping (DTW). You will use GIFs from the GIFGIF+ dataset.

You will analyze the *dynamic* data for this given emotion category. In other words, unlike A2, this time you will analyze sequences over time, as opposed to independent frames.

Dataset

Choose *one* emotion. For example, you may wish to study the affective phenomenon you analyzed in Assignment 1. Download the dataset corresponding to your selected emotion using the API at http://gifgif.media.mit.edu/labs/api .

You may use this notebook to download as many examples as you deem necessary (default: 500).

Tasks

1. Use OpenFace (e.g. in Google CoLab or locally) to extract the features you need (e.g. FACS/gaze/head orientation) from the dataset. You may need to modify the code from the Week 4 Activity or use ffmpeg .

2. Use your knowledge from A2 about **clustering** (K-Means or GMM) using a distance metric that is suited for sequential data (e.g. **multidimensional DTW**). What are some **social signals** that underlie each emotional "category" (e.g. shaking the head, sigh)? Visualize these clusters using your preferred method. In your Jupyter notebook, provide examples of these social signals.

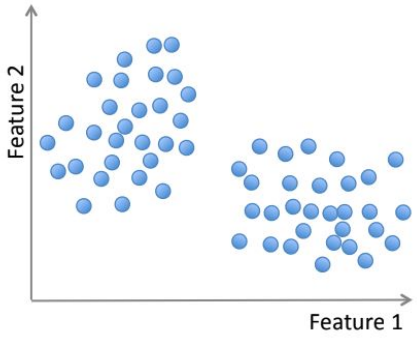
3. Choose **one social signal** that you identified during your clustering process, and create a training (80%) and test set (20%) with at least 30 samples total. Use K-Nearest Neighbours (K=1 is acceptable) and 5-fold cross-validation to report the performance of K-NN and multidimensional DTW on your dataset.

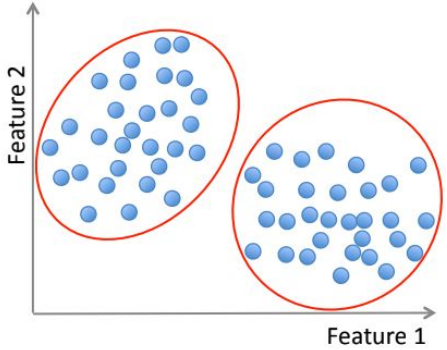
Machine Learning Review

Credit: AI4ALL Introduction to Machine Learning

Unsupervised Learning

- **Unsupervised Learning**: a type of machine learning where we attempt to make inferences about *unlabelled* data

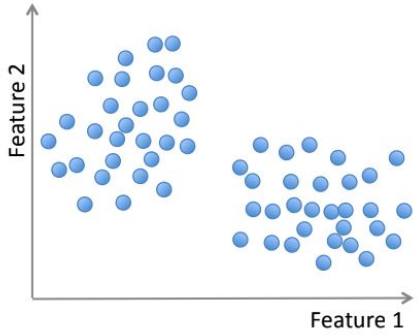


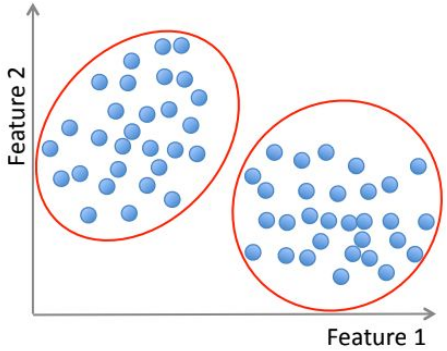


Unsupervised Learning

- **Unsupervised Learning**: a type of machine learning where we attempt to make inferences about *unlabelled* data

- These inferences could be:
  - Different ways to group that data
  - Anomalies that exist in the data
  - Finding new patterns in the data



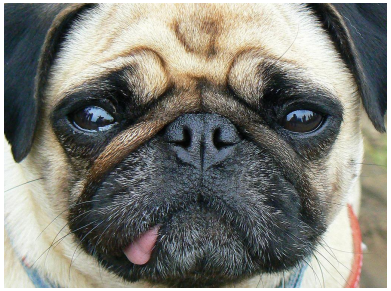


Supervised Learning

- **Supervised Learning**: a type of machine learning where we are given data as an input and attempt to output the correct **label**

- For example, we can take an image as input and assign it with the label *cheetah*, *dog*, *wolf*, etc.

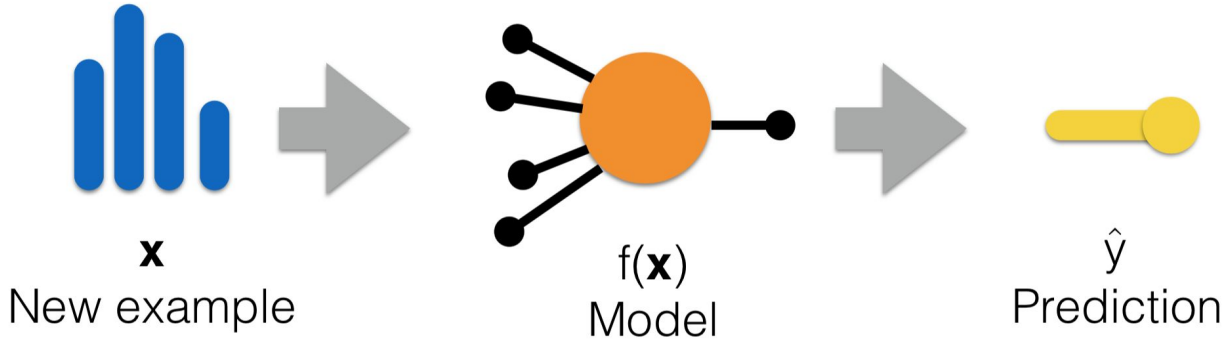
- A specific instance of our data (one particular email or image) is called an **example**



Building Models

- We want to build a **model** of the world, one that understands how to correctly assign a label to an example

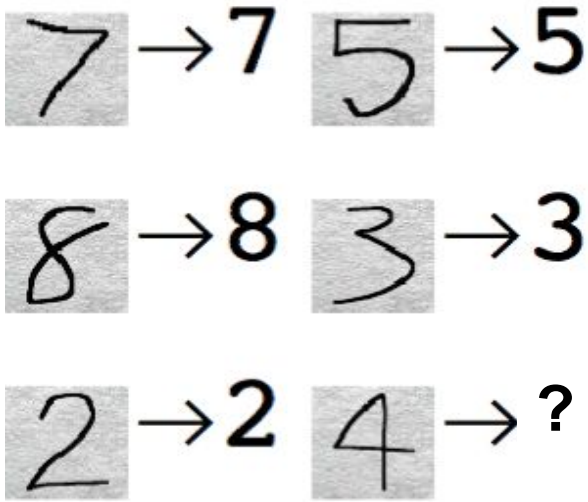
- You can think of a machine learning model as a mathematical function that maps examples to predicted labels



Supervised Learning

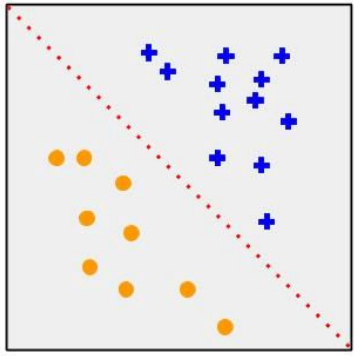
- We learn from examples that are already labelled

- *Supervised* because someone assigns what the correct labels are

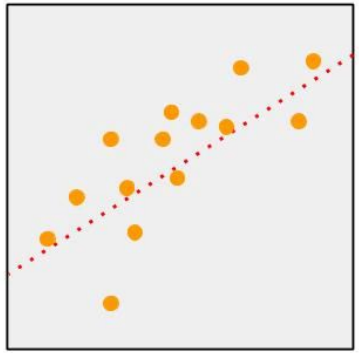


Supervised Learning

- Two types of tasks:
- **Classification**: When the label is a specific *class*
  - Determining if mail is spam or not spam
  - Determining if a picture has a cat or not
- **Regression:** When the label is a *real number*
  - Predicting tomorrow's temperature
  - Predicting the cost of coffee in 2020
- Classification labels are **discrete**
  - Small number of possible values
- Regression labels are **continuous**



Classification



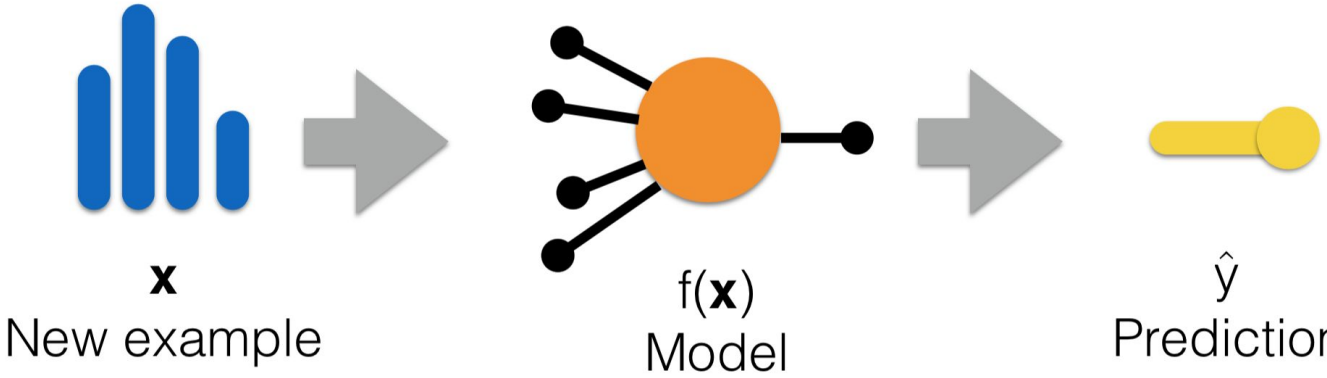
Regression

TRAINING AND GENERALIZATION

Training

- **Training:** the process of using data to improve a model so that it can learn to predict the correct label

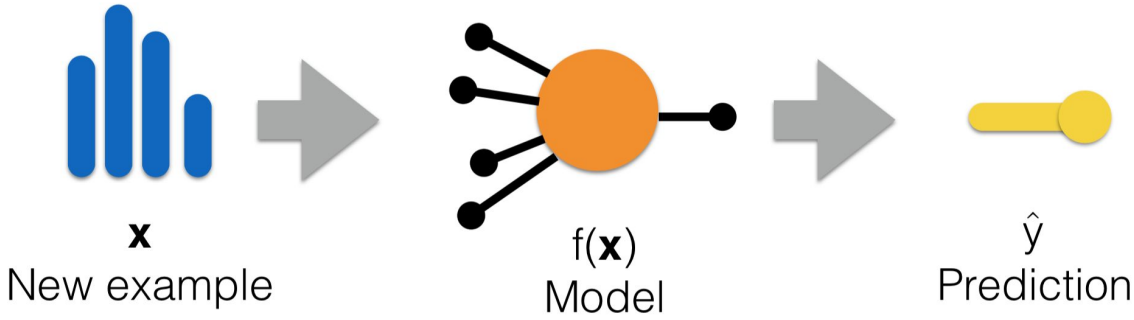
- Different ML algorithms approach training differently



Training Set and Training Error

- To train our model, we use data that we know is correctly labelled -- we call this our **training set**

- We measure our model's performance during training by looking at **training error**, or how accurate its predictions are for the examples we have seen in the training set



Generalization

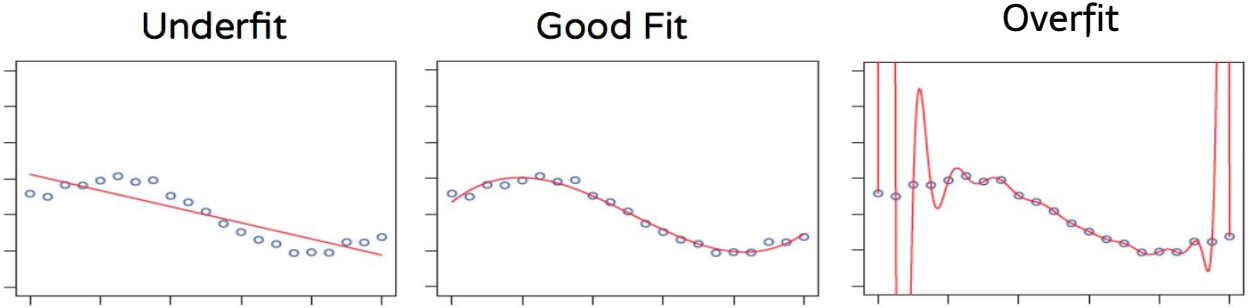
- Doing well on just the training data is not enough
- With training data, we are given the correct labels for each example and can train our model to label those specific examples well
- But how do we know our model will **generalize** well on future, unseen examples?

Fitting to your data

- Just because our model does well on training data does not mean it will do well on unseen data
- **Memorization:** Ability to do well *only* on data we have seen
- **Generalization:** Ability to do well on data we have not seen
- Instead of memorizing the correct labels for the specific examples in our training set, find patterns in the data that allow us to generalize

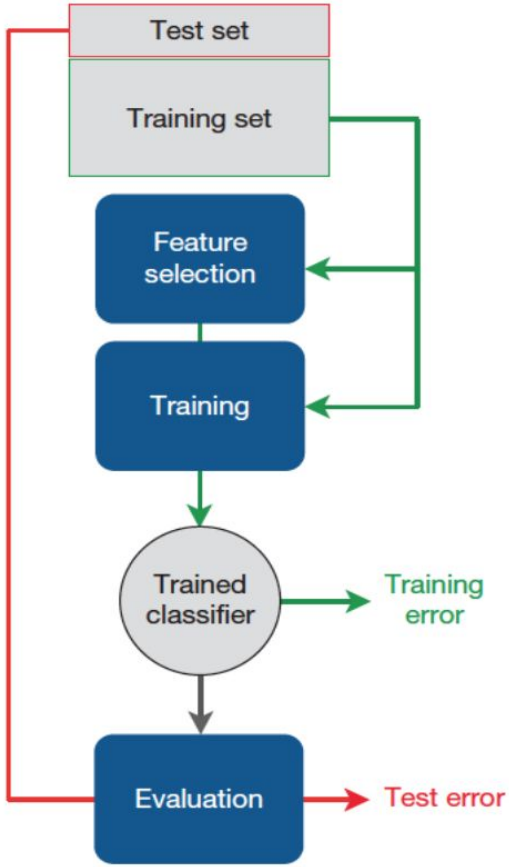
Fitting to your data

- **Overfitting:** When we fit too specifically to the data we've seen (memorization)
- **Underfitting:** When the model doesn't even learn to do well on the data we have seen
- **Good Fit:** If the model understands the patterns in our data and can generalize



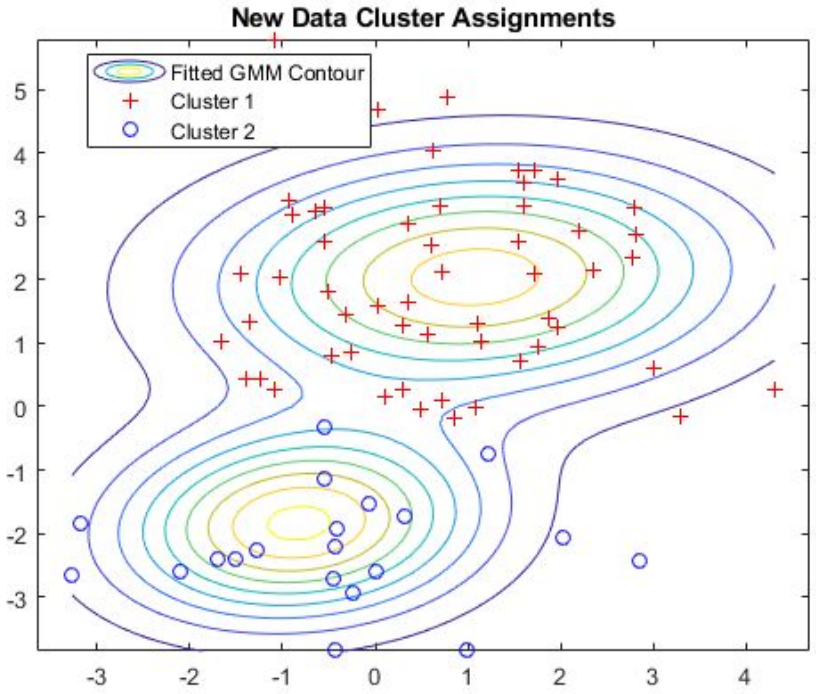
Using a test set

- One way to avoid overfitting is to use a **test set**
- This is a dataset with correct labels that we do not show to our model until *after* it has finished training
- We apply the trained model to the test set to see if it can predict the correct labels on these previously unseen examples
- This can tell us if we've overfit to our training set, and if so, that we should go back and change things in our training process



Small Datasets

- How do you know if your accuracy didn't just happen because you got lucky on your test samples?

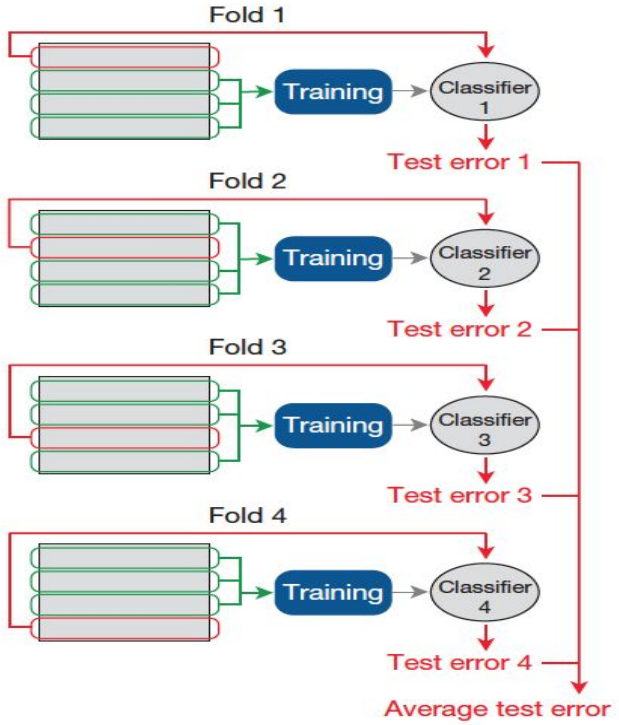


https://towardsdatascience.com/gaussian-mixture-models-d13a5e915c8e

Cross-Validation

- Especially useful if dataset is small
- Break your data down into a training and testing set, e.g. 75-25 split
- For each “fold” we ensure that training and test data is disjoint
- To balance the size of test and training set, we can repeat this many times

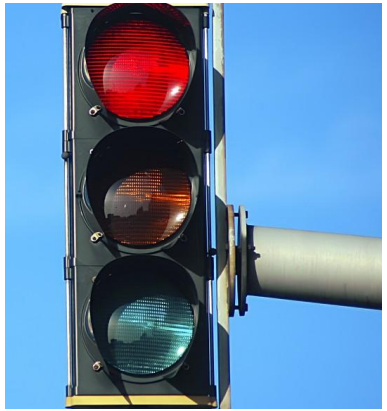
E.g. 4-fold cross-validation



EVALUATION

How can we tell how good our system is?

- One way is to look at the **accuracy** of a model
- For supervised learning:
  - Say we have 100 images of street lights that we want to label as being green, yellow, or red
  - If our model correctly classifies 95 of these 100 images, then we say our model has a 95% accuracy



Other Metrics for Evaluation

- However, accuracy doesn't tell us the whole picture
- There are different types of mistakes that we can make
- Take the example of detecting whether there is a fire in your home or not
- **False Positive (FP)**: When you think there is a fire, but there isn't
- **False Negative (FN)**: When you think there is not a fire, but there is
- Are these errors equally bad?

Other Metrics for Evaluation

	Actual: Yes	**Actual: No**
Predicted: Yes	True Positive (TP)	False Positive (FP)
Predicted: No	False Negative (FN)	True Negative (TN)

This table, when filled in with actual numbers, is called a **confusion matrix**

Evaluation Example

Actual	**Predicted**	**TP/FP/TN/FN?**
Fire	Fire	TP
No Fire	No Fire	TN
Fire	Fire	TP
No Fire	Fire	FP
Fire	Fire	TP
No Fire	Fire	FP
Fire	No Fire	FN
No Fire	Fire	FP
No Fire	No Fire	TN
No Fire	No Fire	TN

Accuracy? **6/10** or **60%**

TP: 3	FP: 3
FN: 1	TN: 3

Evaluation

	Actual: Yes	Actual: No
Predicted: Yes	True Positive (TP)	False Positive (FP)
Predicted: No	False Negative (FN)	True Negative (TN)

- **Precision**: Probability that a positive prediction is correct, TP / (TP + FP)
- **Recall** (or Sensitivity): Probability that an actual positive outcome is predicted correctly, TP / (TP + FN)
- **Specificity**: Probability that an actual negative outcome is predicted correctly, TN / (TN + FP)
- **F1 Score**: Combination of precision and recall, (2 * Precision * Recall) / (Precision + Recall)

Evaluation Example

Precision: TP / (TP + FP) = ?

Recall: TP / (TP + FN) = ?

Specificity: TN / (TN + FP) = ?

F1 Score: (2 * P * R) / (P + R) = ?

Accuracy? **6/10** or **60%**

TP: 3	FP: 3
FN: 1	TN: 3

Evaluation Example

Precision: TP / (TP + FP) = 3/6 = **50%**

Recall: TP / (TP + FN) = 3/4 = **75%**

Specificity: TN / (TN + FP) = 3/6 = **50%**

F1 Score: (2 * P * R) / (P + R) = **60%**

Accuracy? **6/10** or **60%**

TP: 3	FP: 3
FN: 1	TN: 3

Evaluation

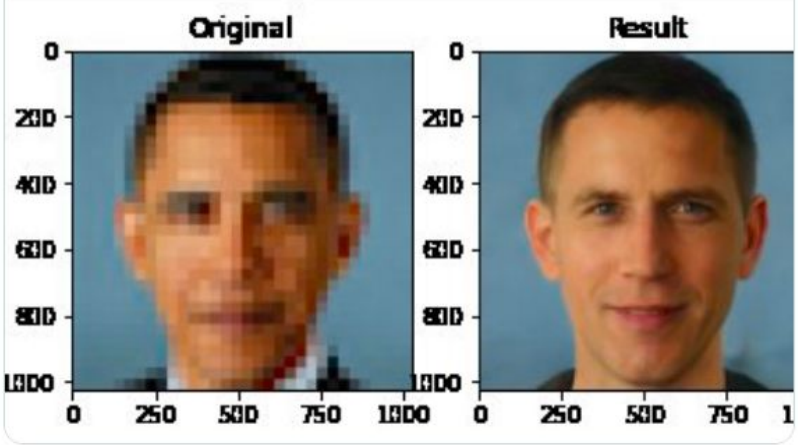
- However, accuracy doesn't tell us the whole picture
- There are different types of mistakes that we can make
- Take the example of detecting whether there is a fire in your home or not
- **False Positive (FP)**: When you think there is a fire, but there isn't
- **False Negative (FN)**: When you think there is not a fire, but there is
- Are these errors equally bad?

Important Considerations

Brad Wyble
@bradpwyble

This image speaks volumes about the dangers of bias in AI

Chicken3gg @Chicken3gg · Jun 20
Replying to @tg_bomze
🤔🤔🤔



9:36 AM · Jun 20, 2020 · Twitter Web App

648 Retweets **1.8K** Likes

Robert Osazuwa Ness @osazuwa · Jun 20

An image of @BarackObama getting upsampled into a white guy is floating around because it illustrates racial bias in #MachineLearning. Just in case you think it isn't real, it is, I got the code working locally. Here is me, and here is @AOC.

Show this thread

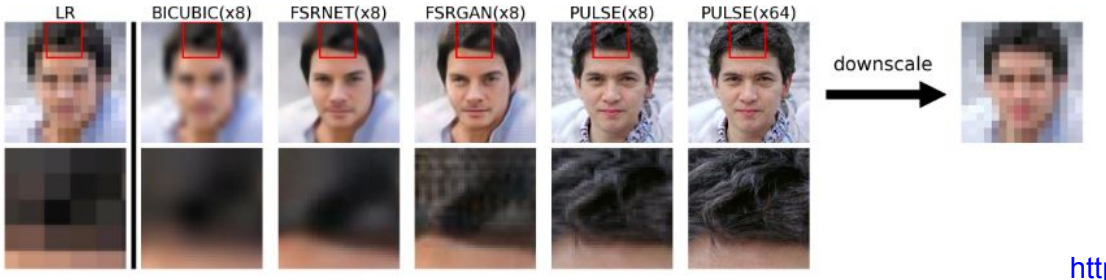
PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models

Code accompanying CVPR'20 paper of the same title. Paper link:
https://drive.google.com/file/d/1fV7FsmunjDuRrsn4KYf2Efwp0FNBtcR4/view

NOTE

We have noticed a lot of concern that PULSE will be used to identify individuals whose faces have been blurred out. We want to emphasize that this is impossible - **PULSE makes imaginary faces of people who do not exist, which should not be confused for real people.** It will **not** help identify or reconstruct the original image.

We also want to address concerns of bias in PULSE. **We have now included a new section in the paper and an accompanying model card directly addressing this bias.**

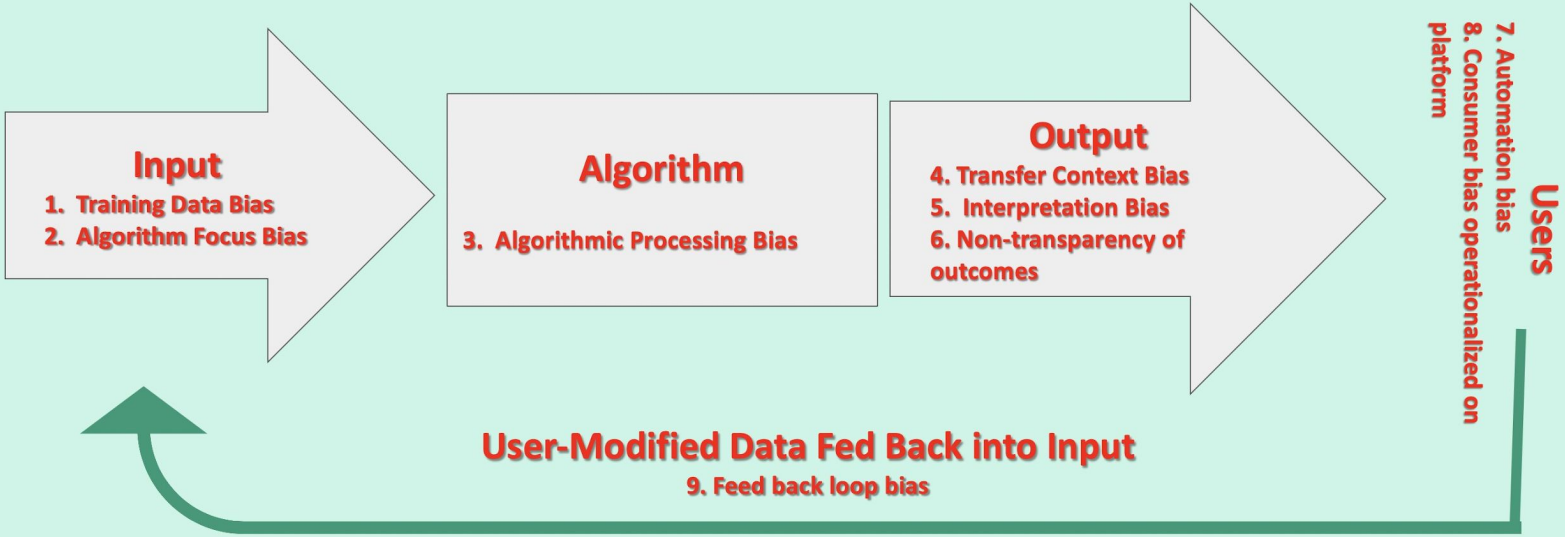


https://github.com/tg-bomze/Face-Depixelizer

This is really bad.

What caused it?

Bias Can Be Introduced in Each Step



Source: Expanded from Danks, D., & London, A. J. (2017).

Model Cards for Model Reporting

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, Timnit Gebru
{mmitchellai,simonewu,andrewzaldivar,parkerbarnes,lucyvasserman,benhutch,espitzer,tgebru}@google.com
deborah.raji@mail.utoronto.ca

Datasheets for Datasets

TIMNIT GEBRU, Google
JAMIE MORGENSTERN, Georgia Institute of Technology
BRIANA VECCHIONE, Cornell University
JENNIFER WORTMAN VAUGHAN, Microsoft Research
HANNA WALLACH, Microsoft Research
HAL DAUMÉ III, Microsoft Research; University of Maryland
KATE CRAWFORD, Microsoft Research; AI Now Institute

Model Cards

Model Card - Smiling Detection in Images

Model Details

- Developed by researchers at Google and the University of Toronto, 2018, v1.
- Convolutional Neural Net.
- Pretrained for face recognition then fine-tuned with cross-entropy loss for binary smiling classification.

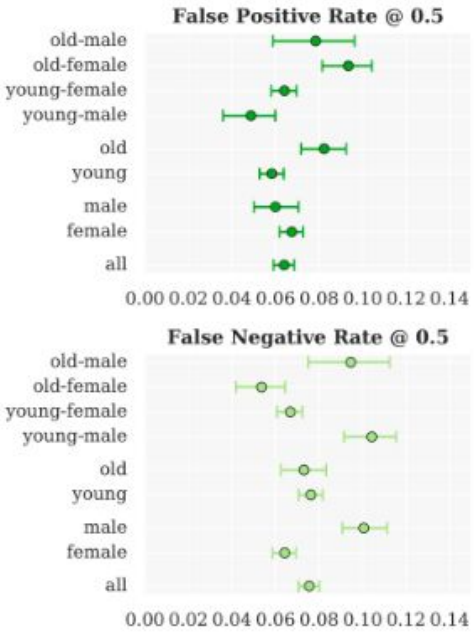
Intended Use

- Intended to be used for fun applications, such as creating cartoon smiles on real images; augmentative applications, such as providing details for people who are blind; or assisting applications such as automatically finding smiling photos.
- Particularly intended for younger audiences.
- Not suitable for emotion detection or determining affect; smiles were annotated based on physical appearance, and not underlying emotions.

Factors

- Based on known problems with computer vision face technology, potential relevant factors include groups for gender, age, race, and Fitzpatrick skin type; hardware factors of camera type and lens type; and environmental factors of lighting and humidity.
- Evaluation factors are gender and age group, as annotated in the publicly available dataset CelebA [36]. Further possible factors not currently available in a public smiling dataset. Gender and age determined by third-party annotators based on visual presentation, following a set of examples of male/female gender and young/old age. Further details available in [36].

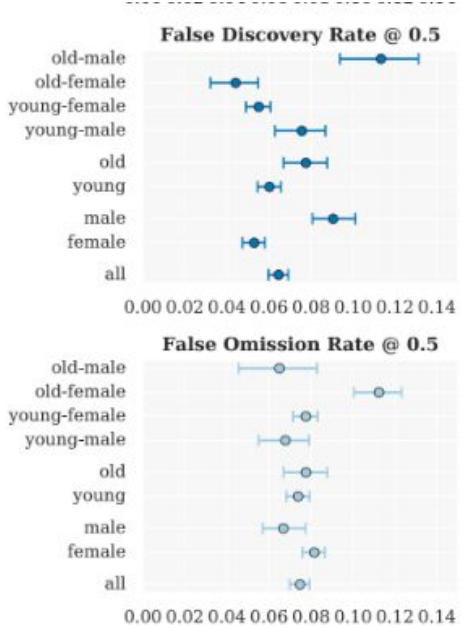
Quantitative Analyses



Model Cards

Metrics

- Evaluation metrics include **False Positive Rate** and **False Negative Rate** to measure disproportionate model performance errors across subgroups. **False Discovery Rate** and **False Omission Rate**, which measure the fraction of negative (not smiling) and positive (smiling) predictions that are incorrectly predicted to be positive and negative, respectively, are also reported. [48]
- Together, these four metrics provide values for different errors that can be calculated from the confusion matrix for binary classification systems.
- These also correspond to metrics in recent definitions of "fairness" in machine learning (cf. [6, 26]), where parity across subgroups for different metrics correspond to different fairness criteria.
- 95% confidence intervals calculated with bootstrap resampling.
- All metrics reported at the .5 decision threshold, where all error types (FPR, FNR, FDR, FOR) are within the same range (0.04 - 0.14).



Training Data

- CelebA [36], training data split.

Evaluation Data

- CelebA [36], test data split.
- Chosen as a basic proof-of-concept.

Ethical Considerations

- Faces and annotations based on public figures (celebrities). No new information is inferred or annotated.

Caveats and Recommendations

- Does not capture race or skin type, which has been reported as a source of disproportionate errors [5].
- Given gender classes are binary (male/not male), which we include as male/female. Further work needed to evaluate across a spectrum of genders.
- An ideal evaluation dataset would additionally include annotations for Fitzpatrick skin type, camera details, and environment (lighting/humidity) details.

Figure 2: Example Model Card for a smile detector trained and evaluated on the CelebA dataset.

Datasheets for Datasets

Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remainder of the questions in this section.

The following steps were taken to process the data:

1. **Gathering raw images:** First the raw images for this dataset were obtained from the Faces in the Wild dataset consisting of images and associated captions gathered from news articles found on the web.
2. **Running the Viola-Jones face detector**[4] The OpenCV version 1.0.0 release 1 implementation of Viola-Jones face detector was used to detect faces in each of these images, using the function cvHaarDetectObjects, with the provided Haar classifier—cascadehaarcascadefrontalfacedefault.xml. The scale factor was set to 1.2, min neighbors was set to 2, and the flag was set to CV HAAR DO CANNY PRUNING.
3. **Manually eliminating false positives:** If a face was detected and the specified region was determined not to be a face (by the operator), or the name of the person with the detected face could not be identified (using step 5 below), the face was omitted from the dataset.
4. **Eliminating duplicate images:** If images were determined to have a common original source photograph, they are defined to be duplicates of each other. An attempt was made to remove all duplicates but a very small number (that were not initially found) might still exist in the dataset. The number of remaining duplicates should be small enough so as not to significantly impact training/testing. The dataset contains distinct images that are not defined to be duplicates but are extremely similar. For example, there are pictures of celebrities that appear to be taken almost at the same time by different photographers from slightly different angles. These images were not removed.
5. **Labeling (naming) the detected people:** The name associated with each person was extracted from the associated news caption. This can be a source of error if the original news caption was incorrect. Photos of the same person were combined into a single group associated with one name. This was a challenging process as photos of some people were associated with multiple names in the news captions (e.g."Bob McNamara" and "Robert McNamara"). In this scenario, an attempt was made to use the most common name. Some people have a single name (e.g. "Madonna" or "Abdullah"). For Chinese and some other Asian names, the common Chinese ordering (family name followed by given name) was used (e.g. "Hu Jintao").
6. **Cropping and rescaling the detected faces:** Each detected region denoting a face was first expanded by 2.2 in each dimension. If the expanded region falls outside of the image, a new image was created by padding the original pixels with black pixels to fill the area outside of the original image. This expanded region was then resized to 250 pixels by 250 pixels using the function cvResize, and cvSetImageROI as necessary. Images were saved in JPEG 2.0 format.
7. **Forming pairs of training and testing pairs for View 1 and View 2 of the dataset:** Each person in the dataset was randomly assigned to a set (with 0.7 probability of being in a training set in View 1 and uniform probability of being in any set in View 2). Matched pairs were formed by picking a person uniformly at random from the set of people who had two or more images in the dataset. Then, two images were drawn uniformly at random from the set of images of each chosen person, repeating the process if the images are identical or if they were already chosen as a matched pair). Mismatched pairs were formed by first choosing two people uniformly at random, repeating the sampling process if the same person was chosen twice. For each chosen person, one image was picked uniformly at random from their set of images. The process is repeated if both images are already contained in a mismatched pair.

Datasheets for Datasets

Collection Process

How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.

The names for each person in the dataset were determined by an operator by looking at the caption associated with the person's photograph.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? How were these mechanisms or procedures validated?

The raw images for this dataset were obtained from the Faces in the Wild database collected by Tamara Berg at Berkeley[3]. The images in this database were gathered from news articles on the web using software to crawl news articles.

Demographic Characteristic	Value
Percentage of female subjects	22.5%
Percentage of male subjects	77.5%
Percentage of White subjects	83.5%
Percentage of Black subjects	8.47%
Percentage of Asian subjects	8.03%
Percentage of people between 0-20 years old	1.57%
Percentage of people between 21-40 years old	31.63%
Percentage of people between 41-60 years old	45.58%
Percentage of people over 61 years old	21.2%

Table 2. Demographic characteristics of the LFW dataset as measured by Han, Hu, and Anil K. Jain. *Age, gender and race estimation from unconstrained face images.* Dept. Comput. Sci. Eng., Michigan State Univ., East Lansing, MI, USA, MSU Tech. Rep.(MSU-CSE-14-5) (2014).

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?

The original Faces in the Wild dataset is a sample of pictures of people appearing in the news on the web. Labeled Faces in the Wild is thus also a sample of images of people found on the news on line. While the intention of the dataset is to have a wide range of demographic (e.g. age, race, ethnicity) and image (e.g. pose, illumination, lighting) characteristics, there are many groups that have few instances (e.g. only 1.57% of the dataset consists of individuals under 20 years old).

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?

Subsequent gender, age and race annotations listed in http://biometrics.cse.msu.edu/Publications/Face/HanJain_UnconstrainedAgeGenderRaceEstimation_MSUTechReport2014.pdf were performed by crowd workers found through Amazon Mechanical Turk.

Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created.

Unknown

Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

Unknown

Does the dataset relate to people? If not, you may skip the remaining questions in this section.

Yes. Each instance is an image of a person.

Project

The goal of the group final project is more than the development of an algorithm. There should be a factor of human evaluation, either at annotation time or evaluation time.

You may choose from the following options:

1) Your own unique idea - You have an insight that would allow you to develop a new research idea. Feel free to talk to the prof or TAs about it! The heft of the project should roughly match the work of those described below. Otherwise, here are some ideas to get you started...

2) Real-time Interaction - Develop an **interactive application** containing a **real-time social signal processing** component (e.g. detect laughter, sigh, side-eye, questions, etc). You will present a video demo of the working scenario. *Examples: Create a virtual agent/robot that interacts with you or multiple people, becomes scared if you say "boo!", laughs when you laugh, comes closer when you beckon, generates affect bursts using a machine learned model, etc. You can consider creating a Unity 3D project, or Pepper Android app.*

- In order to evaluate your social signal processing algorithm's effectiveness, you must create a small **test dataset** (at least 20 examples) and annotate it. Due to limitations in ethics, you may not record external participants in this dataset, but you can record yourselves.
- In the development of the algorithm, you may train using existing datasets if necessary.
- If you plan to recruit participants to do your annotation for you, at least one team member must complete research ethics training (3-4 hours) and present the certificate along with the proposal.
- If your team is comprised of >3 members, consider multimodal signal processing.

Project

3) Offline Social Signal Recognition - Create an **annotated dataset** and **develop a social signal recognition algorithm** to process and recognize the selected social signal(s). Strongly consider choosing data that is dynamic, multimodal, or both. Your dataset must contain at least 200 examples that you must analyze and understand well. For example, you may collect them from YouTube, or expand the annotation of an existing dataset. Due to limitations in ethics, you may not record any new data to create this dataset, and only use publicly available sources. *Examples: create a video dataset of Twitch streamers' reactions, an "awe" dataset, a "thinking" dataset, a contextual dataset of the many meanings of smiles, an audio dataset containing statements vs. questions, an Anime dataset similar to https://www.thesocialiq.com/*

- In order to recruit participants to perform the annotation, at least one member must complete research ethics training (3-4 hours) and present the certificate along with the proposal.
- If your team is comprised of >3 members, consider multimodal signal processing.

4) Social Signal Generation - Develop a machine learning model (e.g. variational autoencoder, GAN, HMM/GMM) to generate social signals. You may choose social signals associated with your Assignment 1 report, or otherwise. The generation of dynamic modalities such as voice (also known as synthesis), music, or video is preferred. *Examples: generate laughter from voice samples*

- You will need to run a user study to show how humans perceive the synthesized work. Therefore, if you select this option, at least one member must complete research ethics training (3-4 hours) and present the certificate along with the proposal.
- If your team is comprised of >3 members, consider multimodal signal processing.

As the project is worth 45% of your final mark, your chosen project must be sized appropriately to the size of your team.