Model-assisted labeling

How to save time and money on data annotation

Olga Petrova, AI Product Manager @ Scaleway

- Who I am:
  - Product Manager for AI PaaS at Scaleway

    Scaleway: European cloud provider, originating from France

- Who I am:
  - Product Manager for AI PaaS at Scaleway

    Scaleway: European cloud provider, originating from France
  - Machine Learning engineer at Scaleway

- Who I am:
  - Product Manager for AI PaaS at Scaleway

    Scaleway: European cloud provider, originating from France
  - Machine Learning engineer at Scaleway
  - Quantum physicist at École Normale Supérieure, the Max Planck

    Institute, Johns Hopkins University

- Who I am:
  - Product Manager for AI PaaS at Scaleway

    Scaleway: European cloud provider, originating from France
  - Machine Learning engineer at Scaleway
  - Quantum physicist at École Normale Supérieure, the Max Planck

    Institute, Johns Hopkins University




www.olgapaints.net

Outline

1. Data annotation: why and how
   Supervised vs. unsupervised ML, structured vs. unstructured data

2. Getting away with less (labeled) data
   Transfer learning
   Semi-supervised learning

3. Model-assisted labeling
   Pre-labeling / Auto-labeling
   Active learning

Outline

1. Data annotation: why and how
   Supervised vs. unsupervised ML, structured vs. unstructured data

2. Getting away with less (labeled) data
   Transfer learning
   Semi-supervised learning

3. Model-assisted labeling
   Pre-labeling / Auto-labeling
   Active learning

Supervised vs. Unsupervised Machine Learning

Supervised vs. Unsupervised Machine Learning

SUPERVISED ML

- For every input, there is an output
- Examples:
  - *Classification*: image → label;
  - *machine translation*: phrase in L1 → phrase in L2;
  - *regression*: vector → value
- Training data is labeled: (input, output) pairs

Supervised vs. Unsupervised Machine Learning

SUPERVISED ML

- For every input, there is an output
- Examples:
  - *Classification*: image → label;
  - *machine translation*: phrase in L1 → phrase in L2;
  - *regression*: vector → value
- Training data is labeled: (input, output) pairs

UNSUPERVISED ML

- *Clustering*: arranging data points in groups
  - Input data points only
- *Generative modeling*: generate new data points from the same distribution
  - Input = noise vector, output is a new data point
- Training data is unlabeled: inputs only

Supervised vs. Unsupervised Machine Learning

SUPERVISED ML

- For every input, there is an output
- Examples:
  - *Classification*: image → label;
  - *machine translation*: phrase in L1 → phrase in L2;
  - *regression*: vector → value
- Training data is labeled: (input, output) pairs

SEMI-SUPERVISED ML

- Supervised ML tasks
- Combination of labeled and unlabeled data
- More on this later

UNSUPERVISED ML

- *Clustering*: arranging data points in groups
  - Input data points only
- *Generative modeling*: generate new data points from the same distribution
  - Input = noise vector, output is a new data point
- Training data is unlabeled: inputs only

Supervised vs. Unsupervised Machine Learning

SUPERVISED ML

- For every input, there is an output
- Examples:
  - *Classification*: image → label;
  - *machine translation*: phrase in L1 → phrase in L2;
  - *regression*: vector → value
- Training data is labeled: (input, output) pairs

UNSUPERVISED ML

- *Clustering*: arranging data points in groups
  - Input data points only
- *Generative modeling*: generate new data points from the same distribution
  - Input = noise vector, output is a new data point
- Training data is unlabeled: inputs only

SEMI-SUPERVISED ML

- Supervised ML tasks
- Combination of labeled and unlabeled data
- More on this later

SELF-SUPERVISED ML

- For every input, there is an auto-generated output
- E.g. *autoencoder*: input = output
- Common in NLP: predictive language modeling

Supervised vs. Unsupervised Machine Learning

SUPERVISED ML

- For every input, there is an output
- Examples:
  - *Classification*: image → label;
  - *machine translation*: phrase in L1 → phrase in L2;
  - *regression*: vector → value
- Training data is labeled: (input, output) pairs

UNSUPERVISED ML

- *Clustering*: arranging data points in groups
  - Input data points only
- *Generative modeling*: generate new data points from the same distribution
  - Input = noise vector, output is a new data point
- Training data is unlabeled: inputs only

SEMI-SUPERVISED ML

- Supervised ML tasks
- Combination of labeled and unlabeled data
- More on this later

SELF-SUPERVISED ML

- For every input, there is an auto-generated output
- E.g. *autoencoder*: input = output
- Common in NLP: predictive language modeling

Structured vs. Unstructured data

STRUCTURED DATA

- Tabulated data

ID	Age	Gender	Weight	Diagnosis
264264	27	0	72	1
908696	61	1	80	0

- No manual annotation

Structured vs. Unstructured data

STRUCTURED DATA

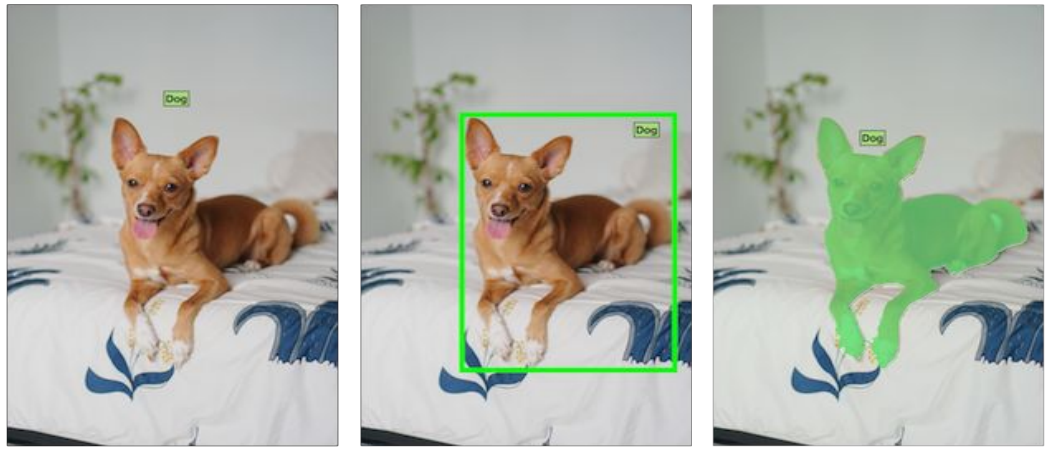
- Tabulated data

ID	Age	Gender	Weight	Diagnosis
264264	27	0	72	1
908696	61	1	80	0

- No manual annotation

UNSTRUCTURED DATA

- Images, videos, text, audio, ...
- Data that is easy to process for humans
  - Image recognition
  - Natural language understanding
  - Etc
- Difficult to automate without ML
- Annotations assigned by hand



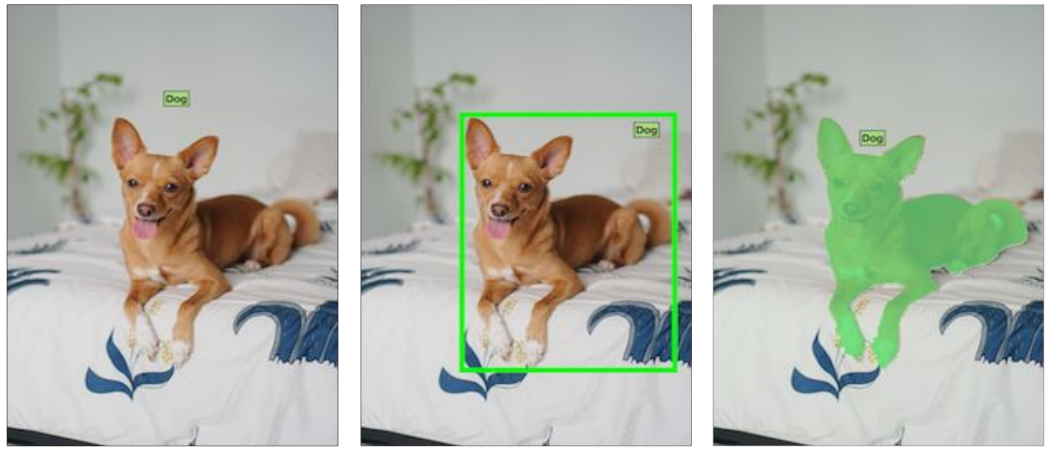
Structured vs. Unstructured data

MANUAL ANNOTATION CHALLENGES

- Domain knowledge
- Difficult to scale, time consuming, expensive
- Multiple annotators per data point
- NLP: language barrier

UNSTRUCTURED DATA

- Images, videos, text, audio, ...
- Data that is easy to process for humans
  - Image recognition
  - Natural language understanding
  - Etc
- Difficult to automate without ML
- Annotations assigned by hand



Outline

1. Data annotation: why and how
   Supervised vs. unsupervised ML, structured vs. unstructured data

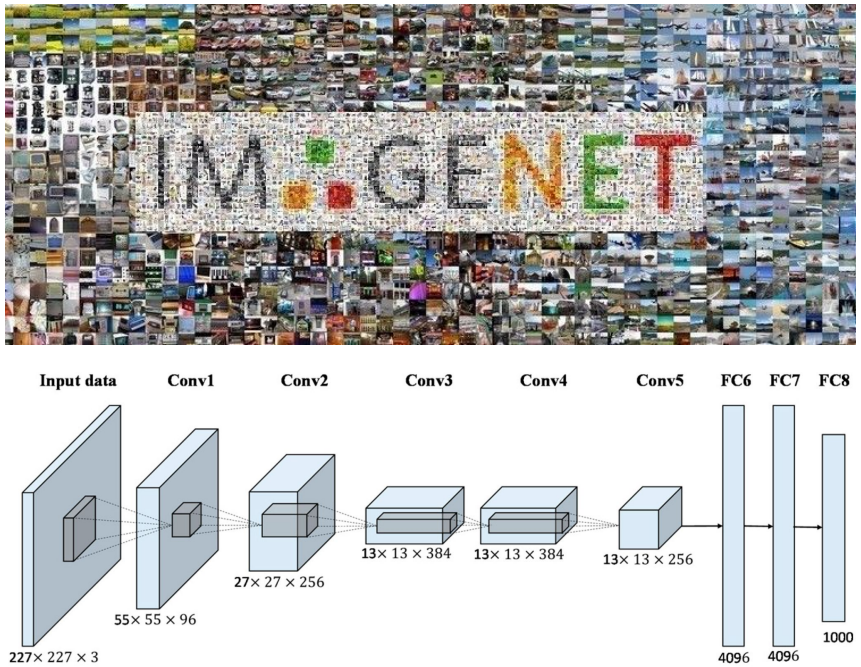
2. Getting away with less (labeled) data
   Transfer learning
   Semi-supervised learning

3. Model-assisted labeling
   Pre-labeling / Auto-labeling
   Active learning

Transfer Learning

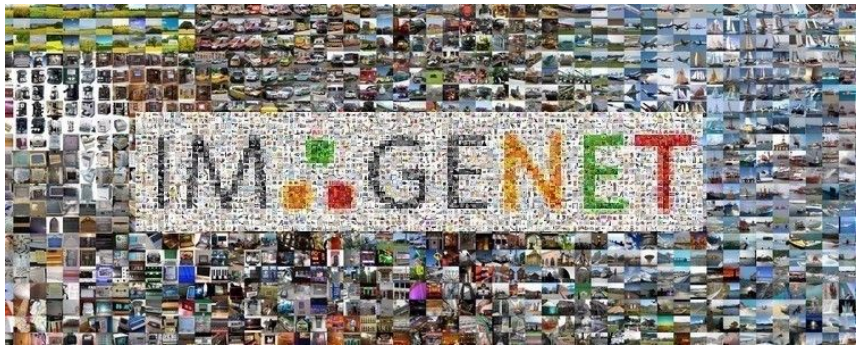
COMPUTER VISION

- Common starting point: model that has been trained on the ImageNet
- Replace outer layer & fine-tune on your data



Transfer Learning

COMPUTER VISION



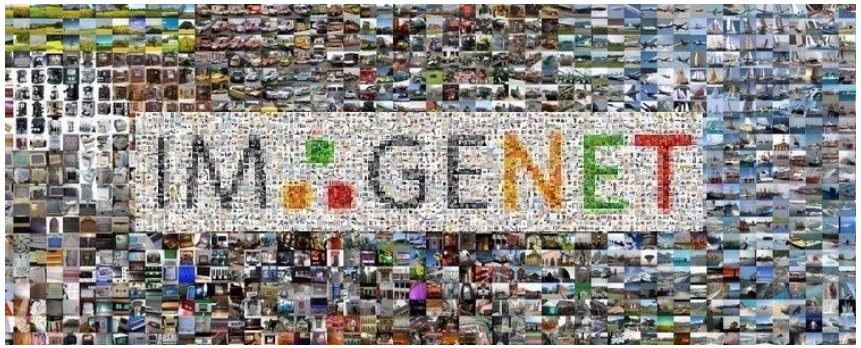
- Common starting point: model that has been trained on the ImageNet
- Replace outer layer & fine-tune on your data

NLP

- Pre-trained self-supervised models
- Word embeddings (e.g. Word2Vec)
- Contextual word embeddings (e.g. BERT)

Transfer Learning

COMPUTER VISION



- Common starting point: model that has been trained on the ImageNet
- Replace outer layer & fine-tune on your data

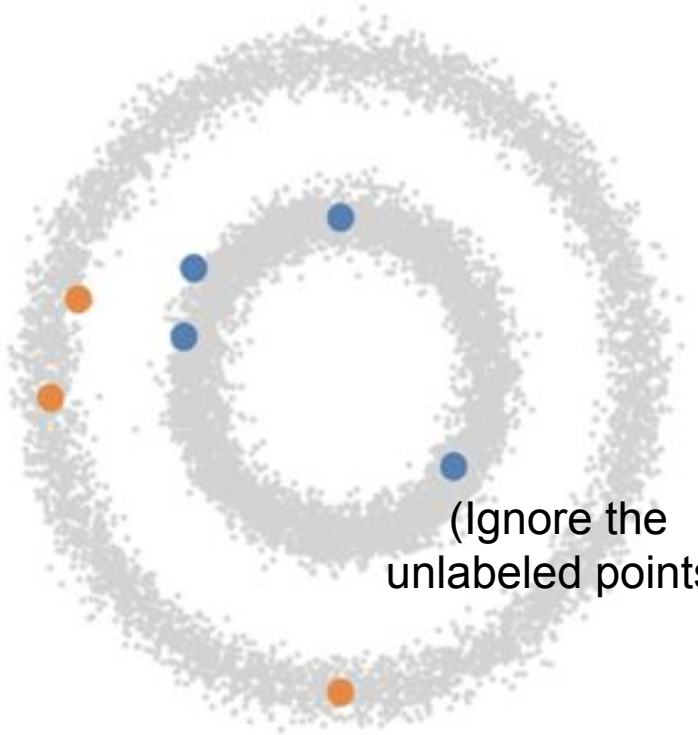
NLP

- Pre-trained self-supervised models
- Word embeddings (e.g. Word2Vec)
- Contextual word embeddings (e.g. BERT)

You can get away with a smaller training set by using a pre-trained model

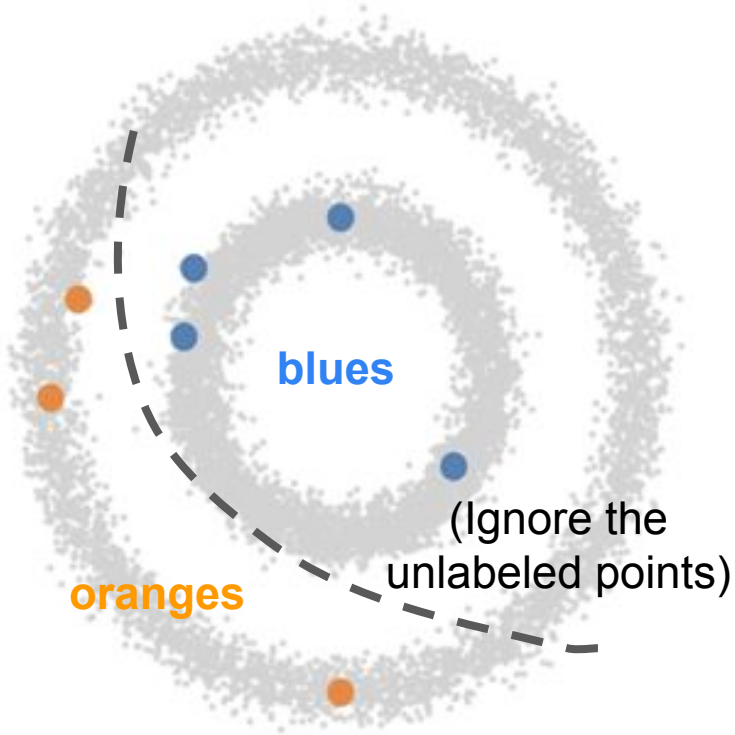
Semi-Supervised Learning

- Goal: use unlabeled data to boost the performance of supervised models
- Binary classification example: **blues** vs. **oranges**
- Draw a decision boundary between classes ignoring the unlabeled points



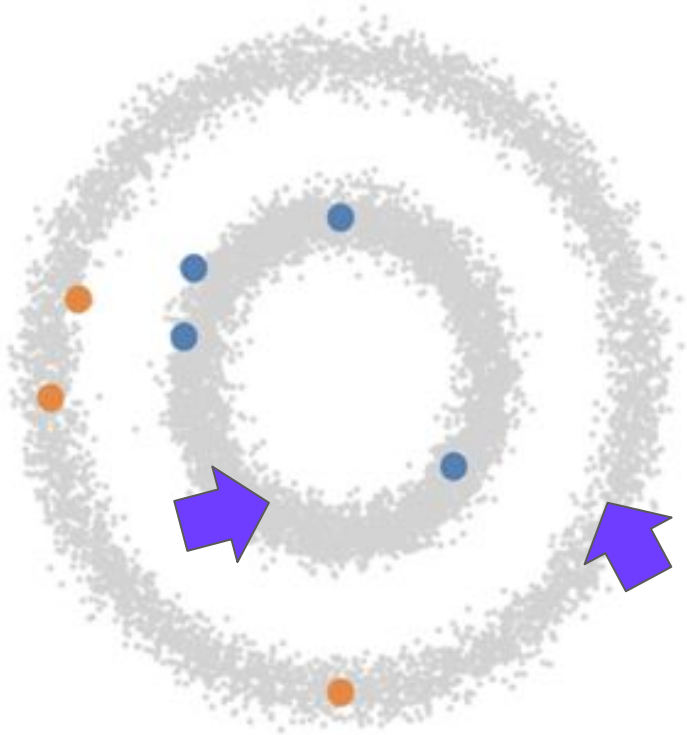
Semi-Supervised Learning

- Goal: use unlabeled data to boost the performance of supervised models
- Binary classification example: **blues** vs. **oranges**
- Draw a decision boundary between classes ignoring the unlabeled points



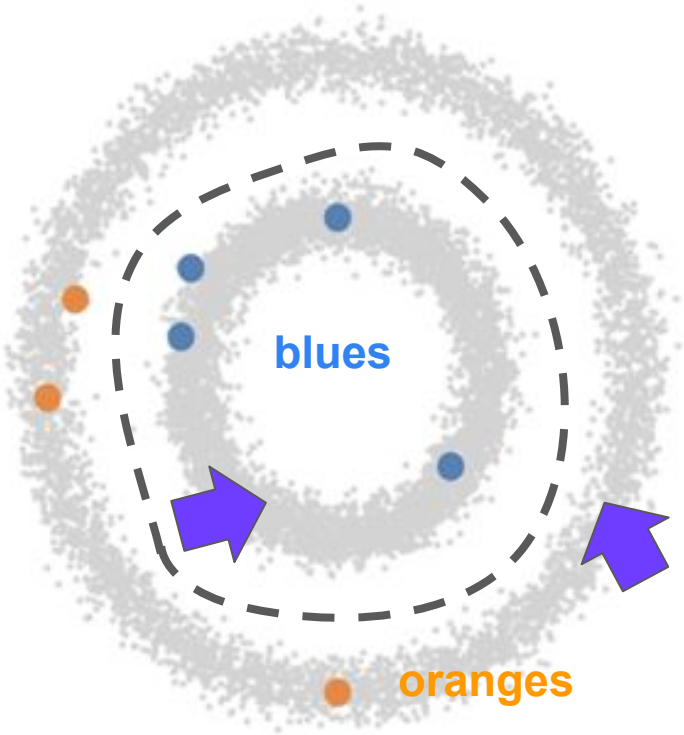
Semi-Supervised Learning

- Goal: use unlabeled data to boost the performance of supervised models
- Binary classification example: **blues** vs. **oranges**
- Draw a decision boundary between classes ignoring the unlabeled points
- What if we take the unlabeled points into account?
- The data can be seen to form two rings



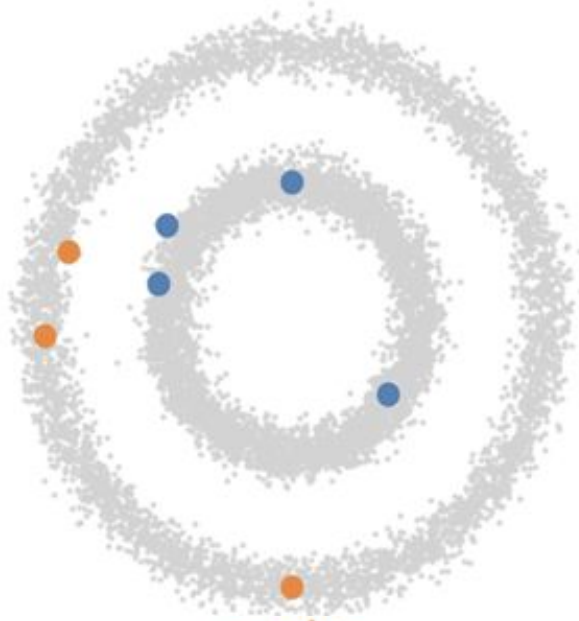
Semi-Supervised Learning

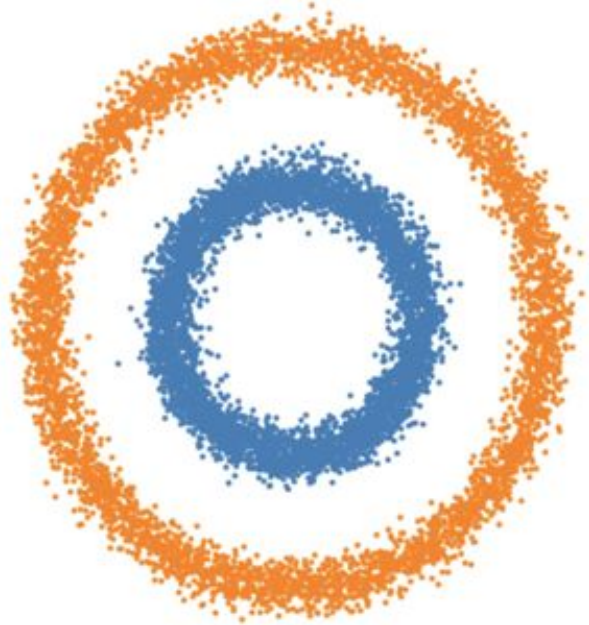
- Goal: use unlabeled data to boost the performance of supervised models
- Binary classification example: **blues** vs. **oranges**
- Draw a decision boundary between classes ignoring the unlabeled points
- What if we take the unlabeled points into account?
- The data can be seen to form two rings
- Using labeled points, we can draw the correct decision boundary now



Semi-Supervised Learning

- Goal: use unlabeled data to boost the performance of supervised models
- Binary classification example: **blues** vs. **oranges**
- Draw a decision boundary between classes ignoring the unlabeled points
- What if we take the unlabeled points into account?
- The data can be seen to form two rings
- Using labeled points, we can draw the correct decision boundary now
- **Unlabeled points give information about the distribution of the data**
- **Labeled points are used to assign class labels**





Outline

1. Data annotation: why and how
   Supervised vs. unsupervised ML, structured vs. unstructured data

2. Getting away with less (labeled) data
   Transfer learning
   Semi-supervised learning

3. Model-assisted labeling
   Pre-labeling / Auto-labeling
   Active learning

Automatic Pre-Labeling

Imagine the following use case:

- You have an online marketplace and you want to classify users' postings with ML (CV and/or NLP)
- You gather and label a huge training dataset and train a classifier

Automatic Pre-Labeling

Imagine the following use case:

- You have an online marketplace and you want to classify users' postings with ML (CV and/or NLP)
- You gather and label a huge training dataset and train a classifier
- But then:
  - A pandemic starts
  - People start selling and buying cute DIY face masks
  - Your model was not trained to recognize them

Automatic Pre-Labeling

Imagine the following use case:

- You have an online marketplace and you want to classify users' postings with ML (CV and/or NLP)
- You gather and label a huge training dataset and train a classifier
- But then:
  - A pandemic starts
  - People start selling and buying cute DIY face masks
  - Your model was not trained to recognize them
- An example of *data drift*
- You need to update your model continuously (fine-tune on new data)

Automatic Pre-Labeling

TRADITIONAL WAY

- Every week/month:
  - randomly choose 5000 new images
  - annotate them
  - evaluate model's performance
  - fine-tune
- Out of 5000, maybe 5% are new data (e.g. face masks)
- The other 4750 labels brought us no value
- How can we avoid this?

Automatic Pre-Labeling

TRADITIONAL WAY

- Every week/month:
  - randomly choose 5000 new images
  - annotate them
  - evaluate model's performance
  - fine-tune
- Out of 5000, maybe 5% are new data (e.g. face masks)
- The other 4750 labels brought us no value
- How can we avoid this?

SMART WAY

- Every week/month:
  - randomly choose 5000 new images
  - get predictions from existing model
  - annotate whatever the model did not recognize
  - fine-tune
- In this example, you only need to label 275 images

Automatic Pre-Labeling

TRADITIONAL WAY

- Every week/month:
  - randomly choose 5000 new images
  - annotate them
  - evaluate model's performance
  - fine-tune
- Out of 5000, maybe 5% are new data (e.g. face masks)
- The other 4750 labels brought us no value
- How can we avoid this?

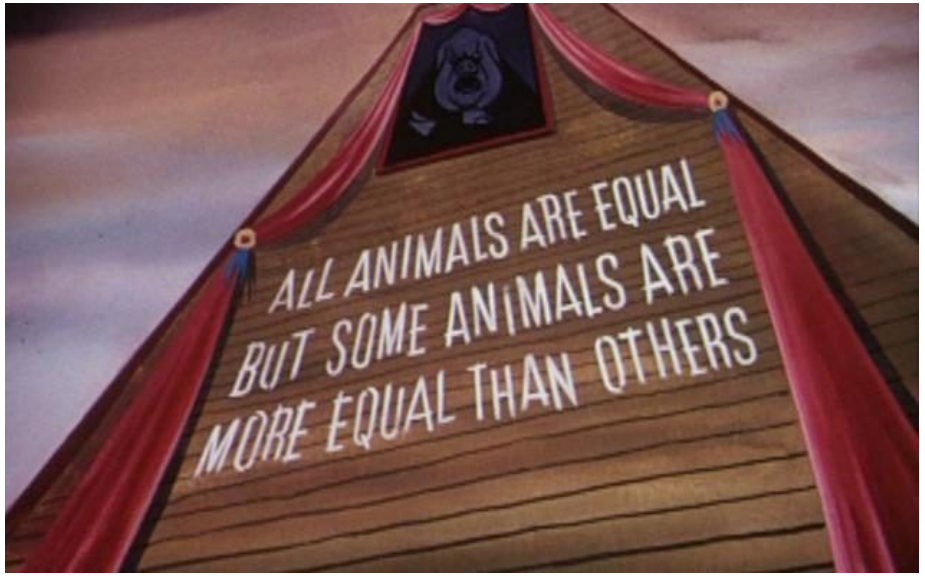
SMART WAY

- Every week/month:
  - randomly choose 5000 new images
  - get predictions from existing model
  - annotate whatever the model did not recognize
  - fine-tune
- In this example, you only need to label 275 images

- Several options:
  - Choose the confidence score threshold
  - Discard or auto-label the recognized images
  - Send high confidence predictions for validation

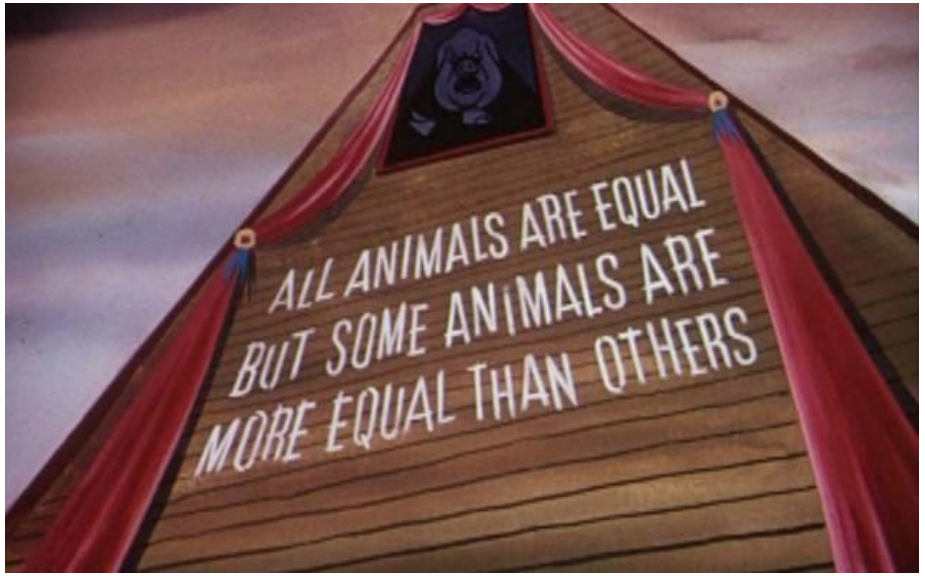
Active Learning

- Active learning: a type of semi-supervised learning
- Goal: select the best subset of data to have labeled
- Not all data is equally useful for model training



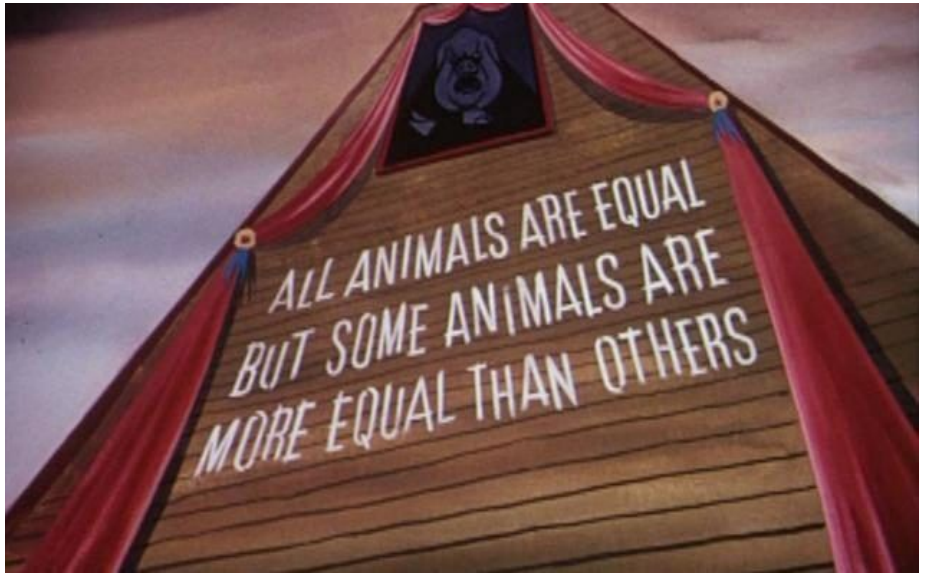
Active Learning

- Active learning: a type of semi-supervised learning
- Goal: select the best subset of data to have labeled
- Not all data is equally useful for model training



Active Learning

- Active learning: a type of semi-supervised learning
- Goal: select the best subset of data to have labeled
- Not all data is equally useful for model training



- Examples of suboptimal labeling choices:
  - groups of similar images

  - noise: out-of-distribution, not representative, etc

Active Learning / Human in the Loop

Active Learning / Human in the Loop

Let’s say we start with an unlabeled dataset ***U***

0. If you don’t have a (pre)trained model, choose a small subset of data, call it ***L***, label it and train the initial model

Active Learning / Human in the Loop

Let's say we start with an unlabeled dataset ***U***

0. If you don't have a (pre)trained model, choose a small subset of data, call it ***L***, label it and train the initial model
1. Get model predictions for ***U***
2. Using a *query strategy*, choose a subset of data to be labeled next

Active Learning / Human in the Loop

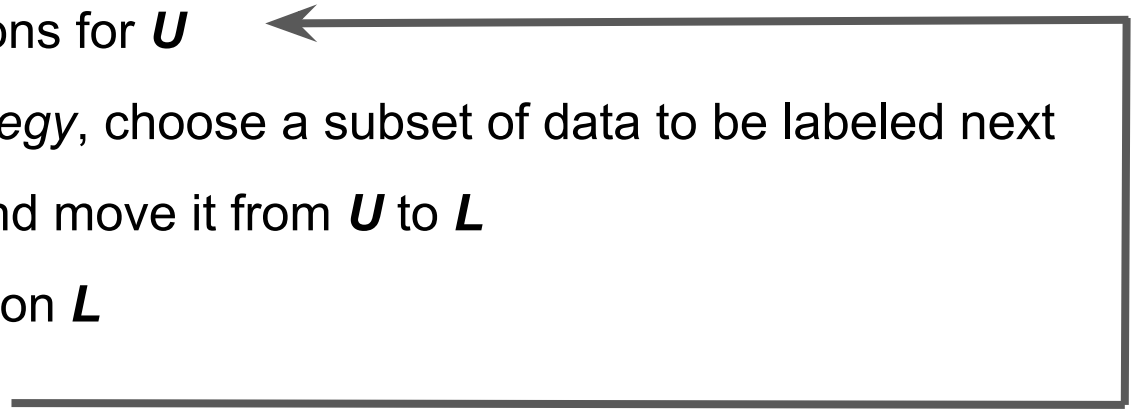
Let's say we start with an unlabeled dataset ***U***

0. If you don't have a (pre)trained model, choose a small subset of data, call it ***L***, label it and train the initial model
1. Get model predictions for ***U***
2. Using a *query strategy*, choose a subset of data to be labeled next
3. Label the subset and move it from ***U*** to ***L***
4. Re-train the model on ***L***

Active Learning / Human in the Loop

Let's say we start with an unlabeled dataset ***U***

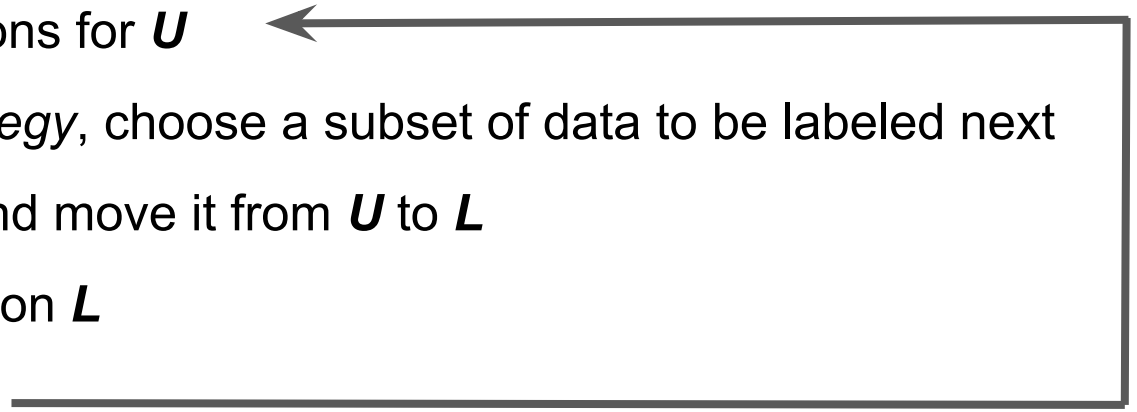
0. If you don't have a (pre)trained model, choose a small subset of data, call it ***L***, label it and train the initial model
1. Get model predictions for ***U***
2. Using a *query strategy*, choose a subset of data to be labeled next
3. Label the subset and move it from ***U*** to ***L***
4. Re-train the model on ***L***
5. Go back to Step 1



Active Learning / Human in the Loop

Let's say we start with an unlabeled dataset ***U***

0. If you don't have a (pre)trained model, choose a small subset of data, call it ***L***, label it and train the initial model
1. Get model predictions for ***U***
2. Using a *query strategy*, choose a subset of data to be labeled next
3. Label the subset and move it from ***U*** to ***L***
4. Re-train the model on ***L***
5. Go back to Step 1



- The idea: at each loop iteration, you label the data that the model can learn from the most
- What is a **query strategy**?

Query Strategies for Classification

LEAST CONFIDENCE

- Choose samples that the model is least sure about

  (= has the most to learn from)

Query Strategies for Classification

LEAST CONFIDENCE

- Choose samples that the model is least sure about
  (= has the most to learn from)

Example: 4 classes (basset, chow-chow, mini poodle, standard poodle)




Query Strategies for Classification

LEAST CONFIDENCE

- Choose samples that the model is least sure about
  (= has the most to learn from)

Example: 4 classes (basset, chow-chow, mini poodle, standard poodle)

Query Strategies for Classification

LEAST CONFIDENCE

- Choose samples that the model is least sure about
  (= has the most to learn from)
- Issue: picks out all the outliers
  - Choose medium, rather than lowest, confidence

Example: 4 classes (basset, chow-chow, mini poodle, standard poodle)




Query Strategies for Classification

LEAST CONFIDENCE

- Choose samples that the model is least sure about (= has the most to learn from)
- Issue: picks out all the outliers
  - Choose medium, rather than lowest, confidence

MARGIN SAMPLING

- Choose samples with the smallest difference between two top choices

Example: 4 classes (basset, chow-chow, mini poodle, standard poodle)




Query Strategies for Classification

LEAST CONFIDENCE

- Choose samples that the model is least sure about (= has the most to learn from)
- Issue: picks out all the outliers
  - Choose medium, rather than lowest, confidence

MARGIN SAMPLING

- Choose samples with the smallest difference between two top choices

Example: 4 classes (basset, chow-chow, mini poodle, standard poodle)

Query Strategies for Classification

LEAST CONFIDENCE

- Choose samples that the model is least sure about (= has the most to learn from)
- Issue: picks out all the outliers
  - Choose medium, rather than lowest, confidence

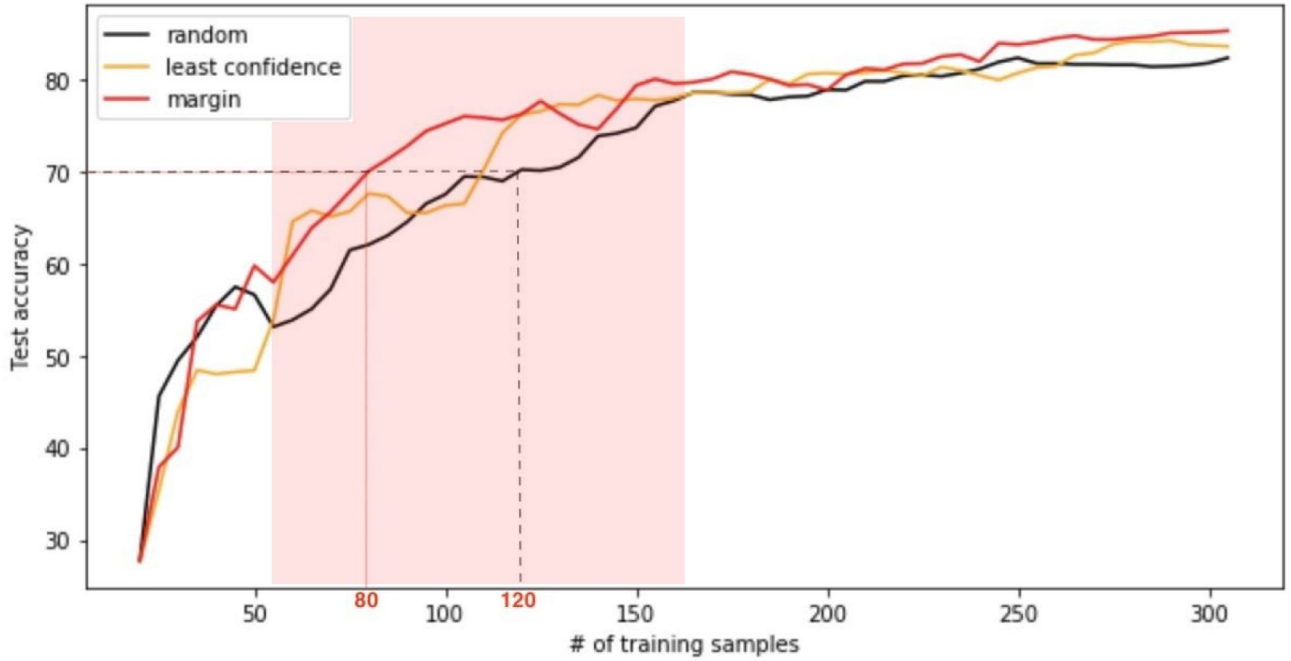
Example: 4 classes (basset, chow-chow, mini poodle, standard poodle)

MARGIN SAMPLING

- Choose samples with the smallest difference between two top choices
- Good for finding decision boundaries between classes
- Binary classification: least confidence = margin sampling




Query Strategies for Classification



Learn more:

blog.scaleway.com/active-learning-some-datapoints-are-more-equal-than-others/

blog.scaleway.com/active-learning-pytorch/

Quick Summary

When on a limited annotation budget, you should:

- Use transfer learning whenever available
- Pre-labeling when you are fine-tuning an existing model
- Active learning if you are training a model from scratch

Quick Summary

When on a limited annotation budget, you should:

- Use transfer learning whenever available
- Pre-labeling when you are fine-tuning an existing model
- Active learning if you are training a model from scratch

Smart Labeling by Scaleway

- Computer Vision annotation platform based on CVAT (Computer Vision Annotation Tool by Intel)
- Currently in free private Beta
- Free Scaleway object storage for up to 75G
- To sign up for the Beta program, email me at opetrova@scaleway.com