CS 181AI
Lecture 4

Training (Part 2)

Arthi Padmanabhan

Jan 25, 2023

Logistics

- Reminder: Assignment 1 is due today at 5pm
- I have office hours 4-5pm (office hours are now posted on webpage)

Today’s Plan

- Group A leads reading discussion
- Debugging low accuracy when training a neural network

Course Path: The First Half

Intro to ML models and hands-on experience working with them

- Run a pre-trained model over your own images (1/23)
- Learn about training process and train a small model on a common dataset (1/25)
- Debug common issues with low accuracy when training (1/30)
- Learn about different types of models and their training processes (2/1)

Ethical AI

- Assess goals in building models
- Assess methods of data collection
- Study how various types of bias affect model results
- Study real use-cases

Computational Resources to Run and Train Models

- Learn about properties of machines that are particularly good for ML
- Study state-of-the-art ML devices + some promising future computing methods (e.g., quantum)
- Learn about how using multiple machines can enable scaling of ML tasks
- Study methods for lowering computational resources needed

Other Resources to Run and Train Models: Memory, Energy

- Compare memory and energy usage of running and training various types of models
- Study strategies for lowering memory and energy usage
- Case study: ML for monitoring agriculture

Activity #1

- Open lec4.ipynb from Lecture 4 Files – this is similar to what we built last time but uses a different dataset (CIFAR instead of MNIST)
- Write a function, evaluate_model(), to compute your model's accuracy
- Save your model's loss and accuracy after each batch over 5 epochs,
- What relationship do you expect loss and accuracy to have? How do your results compare?

Improving Model Accuracy

- Common culprits:
  - Data
  - Model
  - Training Process

Improving Model Accuracy: Data

- ML work often focuses on models because they're more interesting
- Data is often the bane of ML engineer's existence

Data in torchvision datasets

Data in production

Improving Model Accuracy: Data

- Creating good training data
  - Sampling
  - Labeling
  - Dealing with Class Imbalance
  - Dealing with Lack of Sufficient Data

Improving Model Accuracy: Data

- Creating good training data
  - Why sampling?
    - You don’t have access to all the real-world data
    - There’s too much and it would cost too much or be too time-consuming
  - Understanding different sampling methods can help reduce bias

Sampling: Activity #2

- Exercise in groups (5 min): you want to build a model to classify whether a tweet spreads misinformation
  - There are 10M tweets from 10k users over the last 24 months
  - You estimate that 1% of tweets are misinformation
- How would you sample 100K tweets to label and train on?
- You get 100K tweets from 20 annotators and want to look at some labels to estimate quality – how do you decide which + how many labels to look at?
- Imagine you have a stream of an unknown number of tweets coming in (live) and you only want to train on 10k (for memory reasons). How do you decide which 10k to sample such that each tweet has an equal probability of being selected?

Sampling

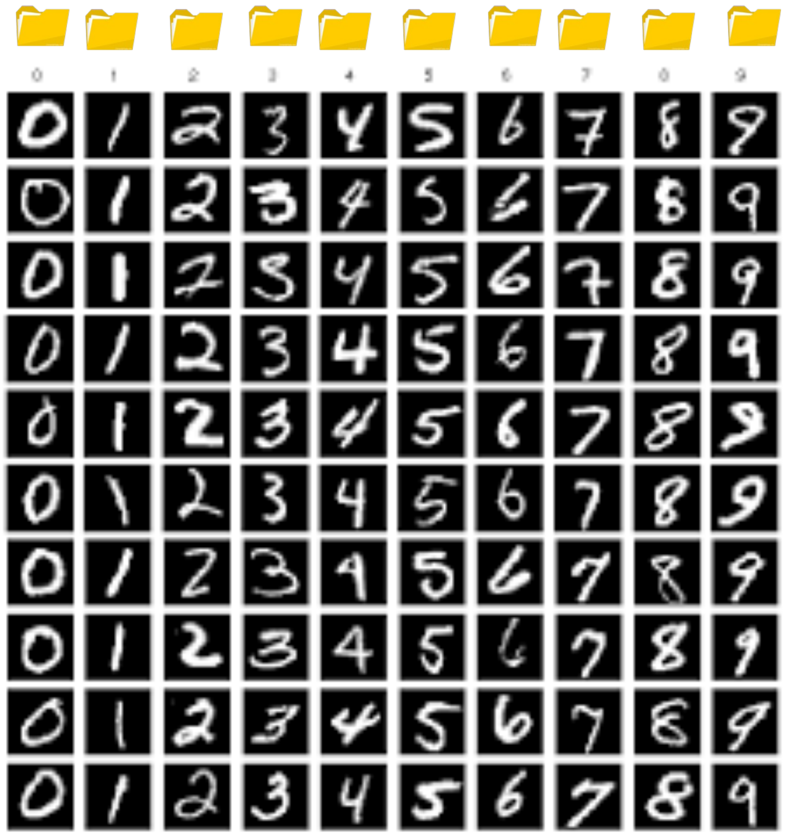
- Non-probability sampling
  - Convenience
  - Snowball
  - Judgment
  - Quota

Sampling

- Probability sampling
  - Simple Random
  - Stratified
  - Weighted
  - Importance
  - Reservoir

Labeling

- When we looked at MNIST, actual dataset looked like this:



Can we always hand label all our data?

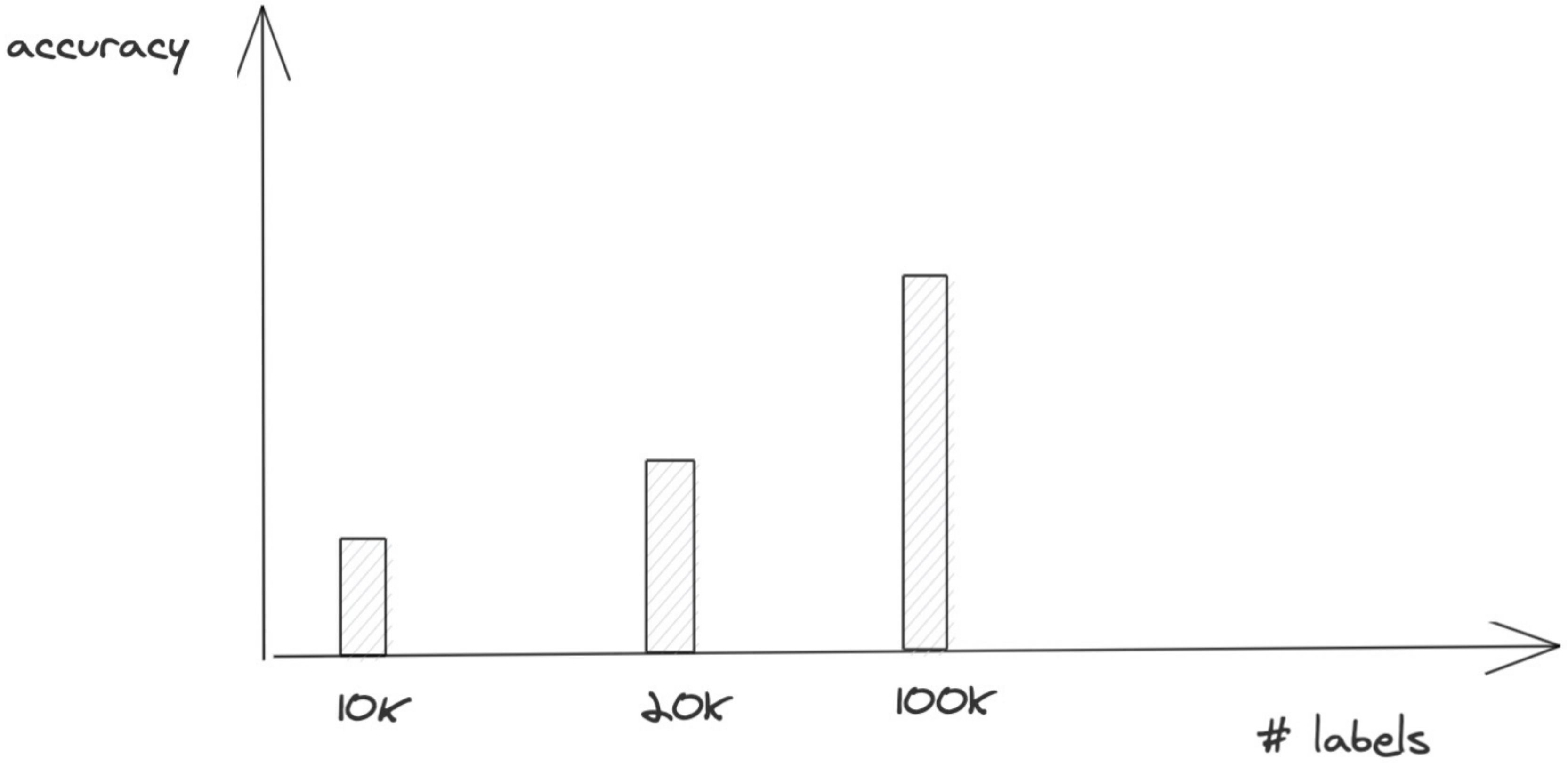
Labeling Example

- How many entities are in the following sentence?

Darth Sidious, known simply as the Emperor, was a Dark Lord of the Sith who reigned over the galaxy as Galactic Emperor of the First Galactic Empire.

Labeling

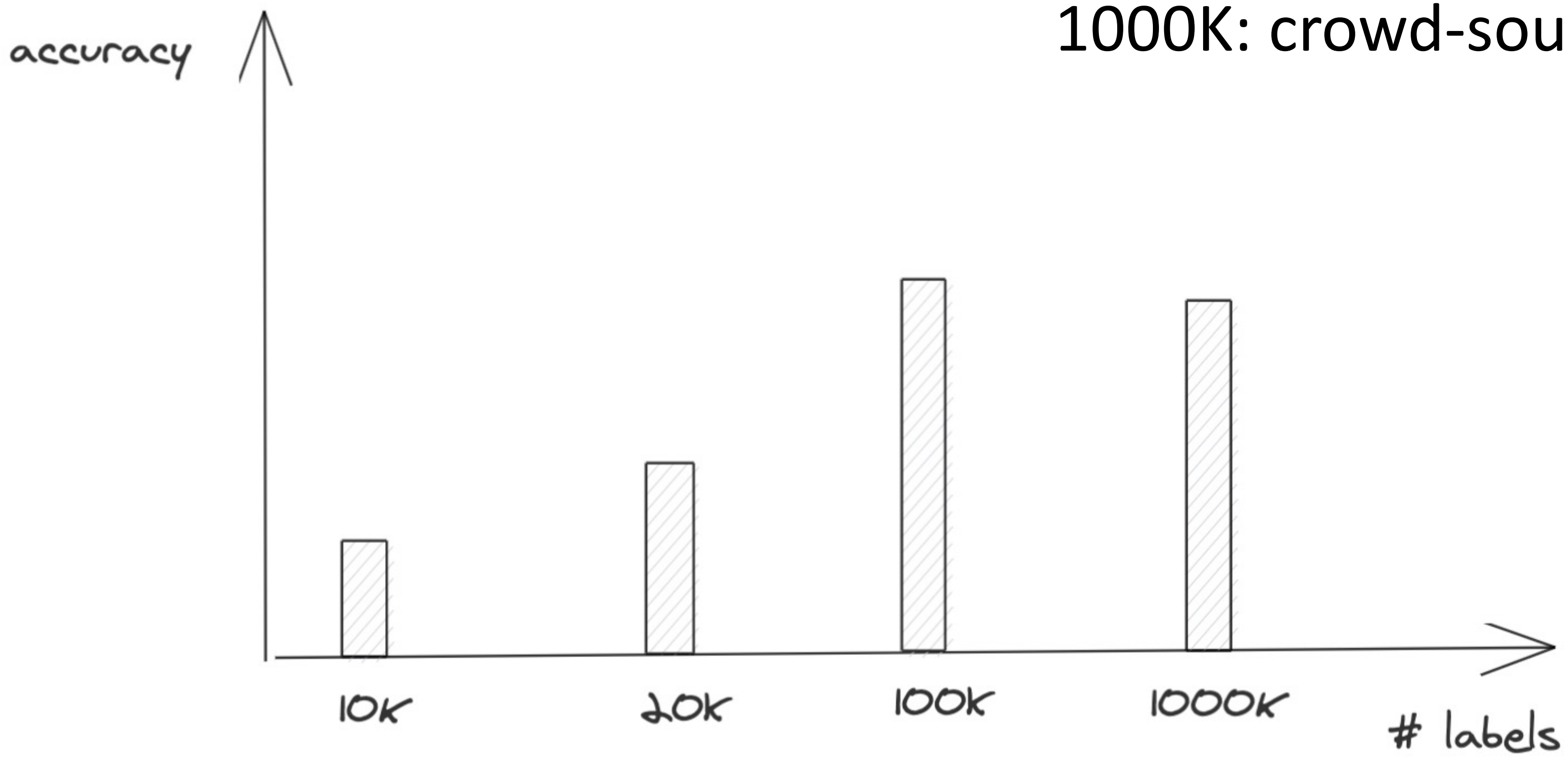
- More data isn't always better



Labeling

- More data isn't always better

100K labels: internal, hand-labeled
1000K: crowd-sourced, noisy



Labeling

- Programmatic Labeling can be messy but much faster and more adaptive
- Semi-supervision, or self-training: start with initial set of labels and train on new samples - only include the ones with a high score
- Active learning – only label those that model is uncertain about

Class Imbalance

- Which model do you hope your bank uses?

Model A	Actual Fraud	Actual Normal
Predicted Fraud	10	10
Predicted Normal	90	890

Model B	Actual Fraud	Actual Normal
Predicted Fraud	90	90
Predicted Normal	10	810

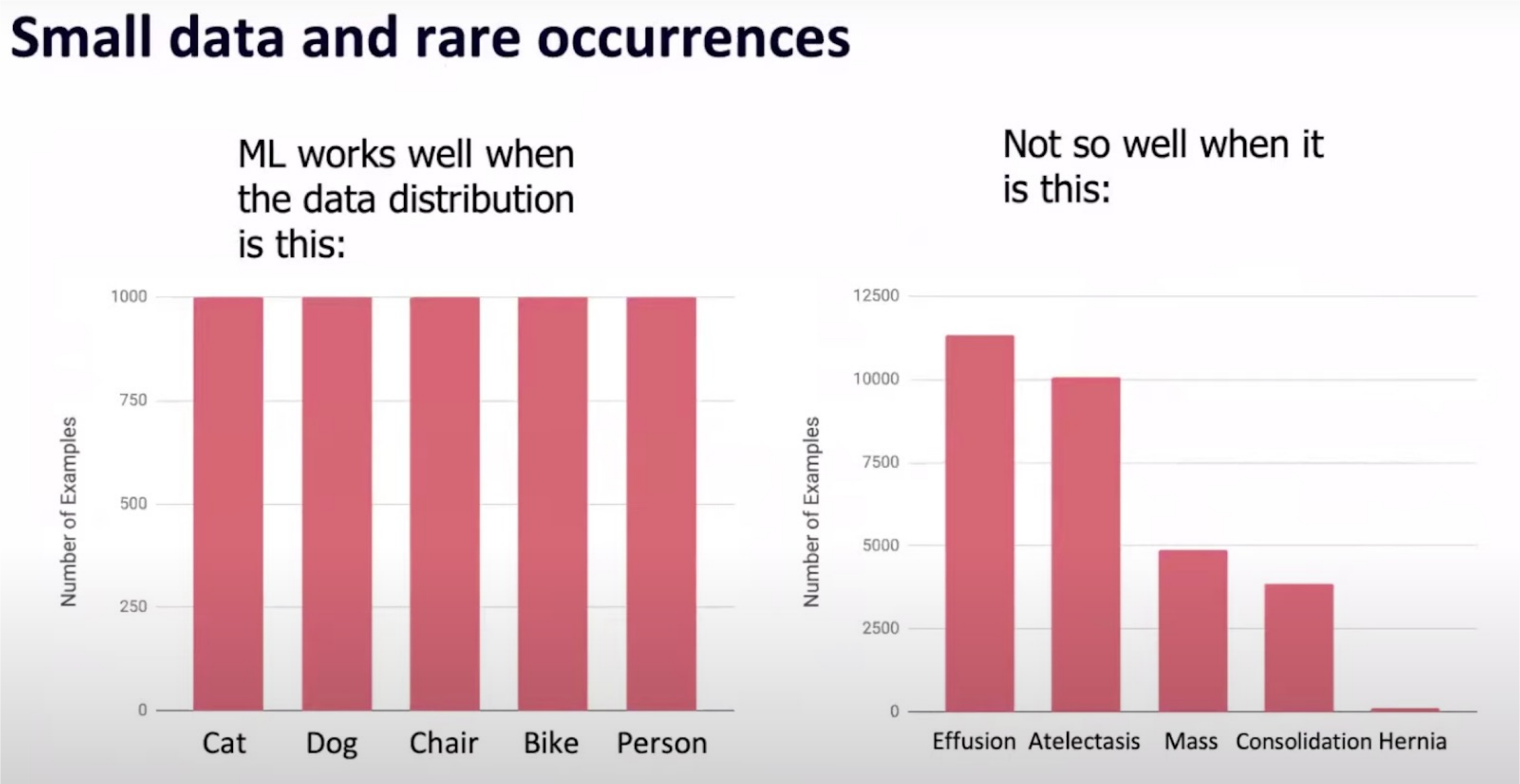
Class Imbalance

- Which model do you hope your bank uses?
  - Just going by accuracy, they both have an accuracy of 90%

Model A	Actual Fraud	Actual Normal
Predicted Fraud	10	10
Predicted Normal	90	890

Model B	Actual Fraud	Actual Normal
Predicted Fraud	90	90
Predicted Normal	10	810

Class Imbalance

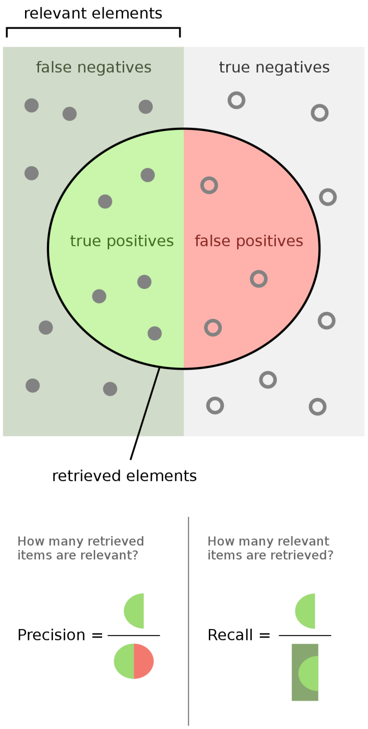


Class Imbalance

- Statistically, predicting the majority class has a higher chance of being right
- Asymmetric cost of error: different cost of wrong prediction

Class Imbalance

- Overall accuracy is most commonly used, but it's insufficient when there's class imbalance because it treats all classes the same
- F1: 2*(precision)*(recall) / (precision + recall)



Class Imbalance

- Precision, Recall, and F1?

Model A	Actual Fraud	Actual Normal
Predicted Fraud	10	10
Predicted Normal	90	890

Model B	Actual Fraud	Actual Normal
Predicted Fraud	90	90
Predicted Normal	10	810

Class Imbalance

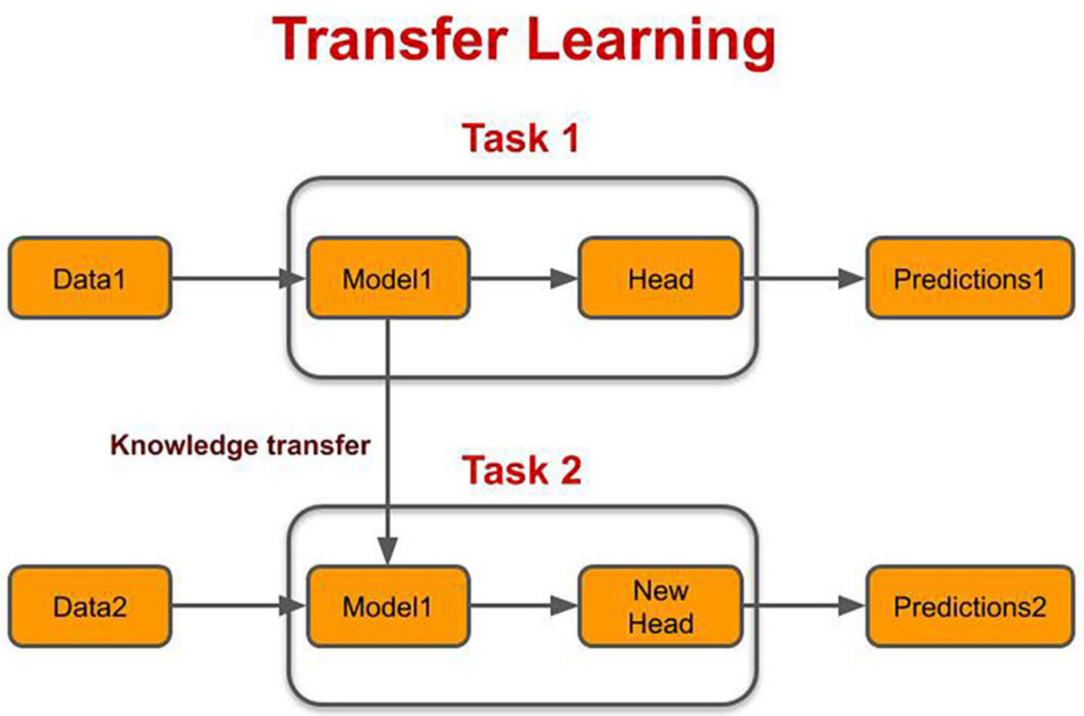
- Model A:
  - Precision = 50% (10/20)
  - Recall = 10% (10/100)
  - F1 = 17%
- Model B:
  - Precision = 50% (90/180)
  - Recall = 90% (90/100)
  - F1 = 64%

Model A	Actual Fraud	Actual Normal
Predicted Fraud	10	10
Predicted Normal	90	890

Model B	Actual Fraud	Actual Normal
Predicted Fraud	90	90
Predicted Normal	10	810

Not Enough Data

- Transfer Learning



Models

- Layers -> generally more will give higher accuracy
- Details of how model structure (i.e., different layer types, order) affects accuracy is beyond scope of this course

Training Process

- Learning rate
- Optimizer
- Common mistakes: eval() vs. train(), didn’t zero the gradients, didn’t pass the correct model’s parameters to the optimizer

Next Time

- Expanding to different types of models