Machine Learning Systems Design

Lecture 4: Feature Engineering

Zoom poll: How is the class going?

- Too fast
- Too slow
- It's going great
- Meh

Reply in chat: what concepts you hope to learn in class?

Stanford University

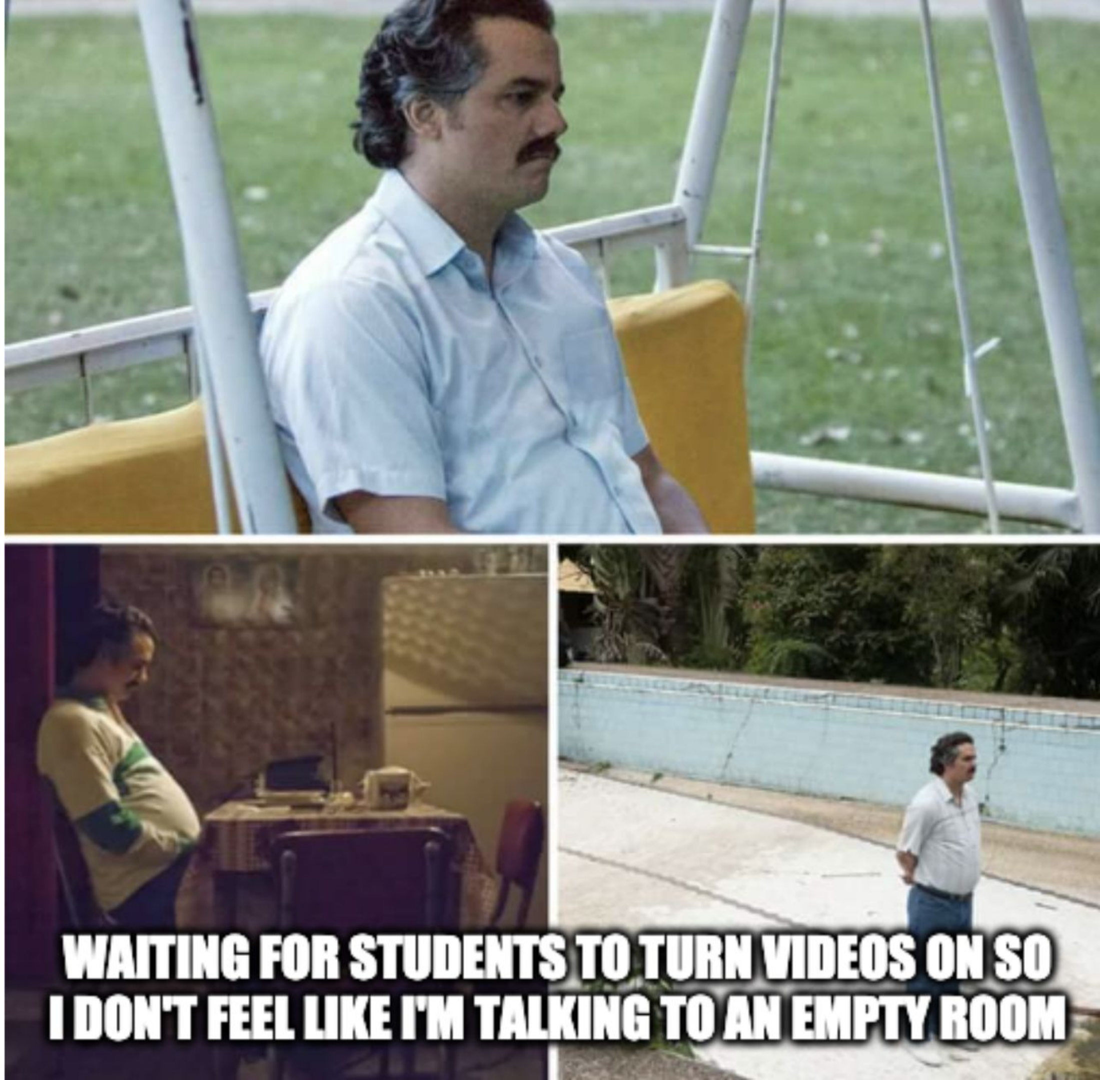
Logistics

- [Team search is happening]

Zoom etiquettes

We appreciate it
if you keep videos on!

- More visual feedback for us to adjust materials
- Better learning environment
- Better sense of who you're with in class!



Agenda

0. Class imbalance (ctd.)
1. Data augmentation
2. Learned features vs. engineered features
3. Breakout exercise
4. Common feature engineering ops
5. Data leakage

Lecture note is on course website / syllabus

Class imbalance (ctd.)

Class imbalance is the norm

- Fraud detection
- Spam detection
- Disease screening
- Churn prediction
- Resume screening
  - E.g. 2% of resumes pass screening
- Object detection
  - Most bounding boxes don't contain any object

People are more interested in unusual/potentially catastrophic events

Image from PyImageSearch

Sources of class imbalance

- Sampling biases
  - Narrow geographical areas (self-driving cars)
  - Selection biases
- Domain specific
  - Costly, slow, or infeasible to collect data of certain classes
- Labeling errors

How to deal with class imbalance

1. Choose the right metrics
2. Data-level methods
3. Algorithm-level methods

1. Choose the right metrics

Model A vs. Model B confusion matrices

Zoom poll:
Which model would you choose?

Model A	**Actual CANCER**	**Actual NORMAL**
Predicted CANCER	10	10
Predicted NORMAL	90	890

Model B	**Actual CANCER**	**Actual NORMAL**
Predicted CANCER	90	90
Predicted NORMAL	10	810

Choose the right metrics

Model A vs. Model B confusion matrices

Model B has a better chance of telling if you have cancer

Model A	**Actual CANCER**	**Actual NORMAL**
Predicted CANCER	10	10
Predicted NORMAL	90	890

Model B	**Actual CANCER**	**Actual NORMAL**
Predicted CANCER	90	90
Predicted NORMAL	10	810

Both have the same accuracy: 90%

Symmetric metrics vs. asymmetric metrics

Symmetric metrics	Asymmetric metrics
Treat all classes the same	Measures a model's performance w.r.t to a class
Accuracy	F1, recall, precision, ROC

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

$$F_1\text{–score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

- TP: True positives
- TN: True negatives
- FP: False positives
- FN: False negatives

Class imbalance: asymmetric metrics

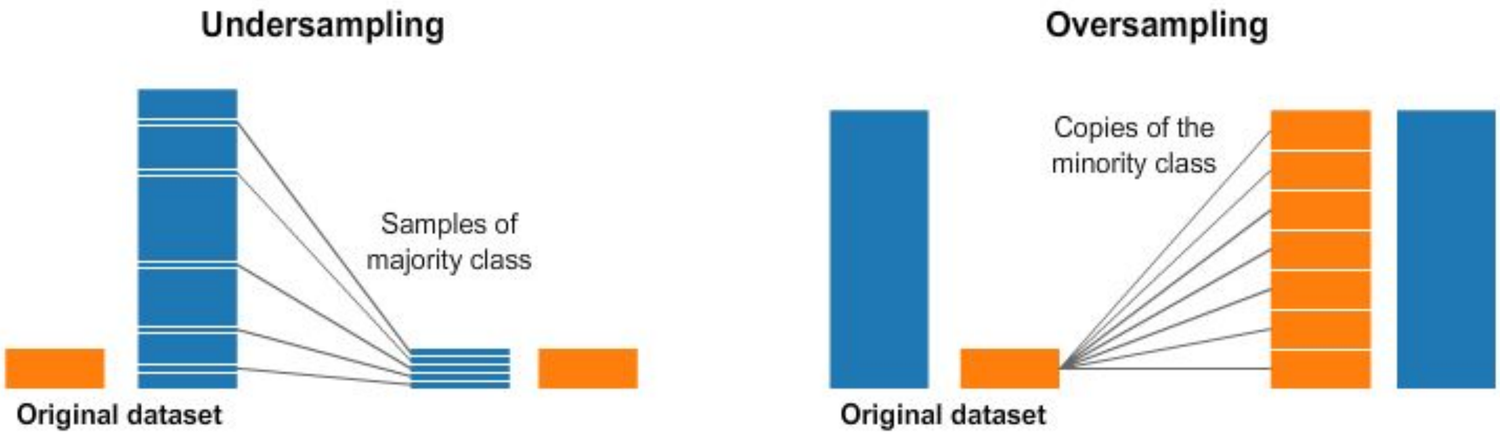
- Your model's performance w.r.t to a class

	CANCER (1)	NORMAL (0)	Accuracy	Precision	Recall	F1
Model A	10/100	890/900	0.9	0.5	0.1	0.17
Model B	90/100	810/900	0.9	0.5	0.9	0.64

⚠️ F1 score for CANCER as 1 is different from F1 score for NORMAL as 1 ⚠️

2. Data-level methods: Resampling

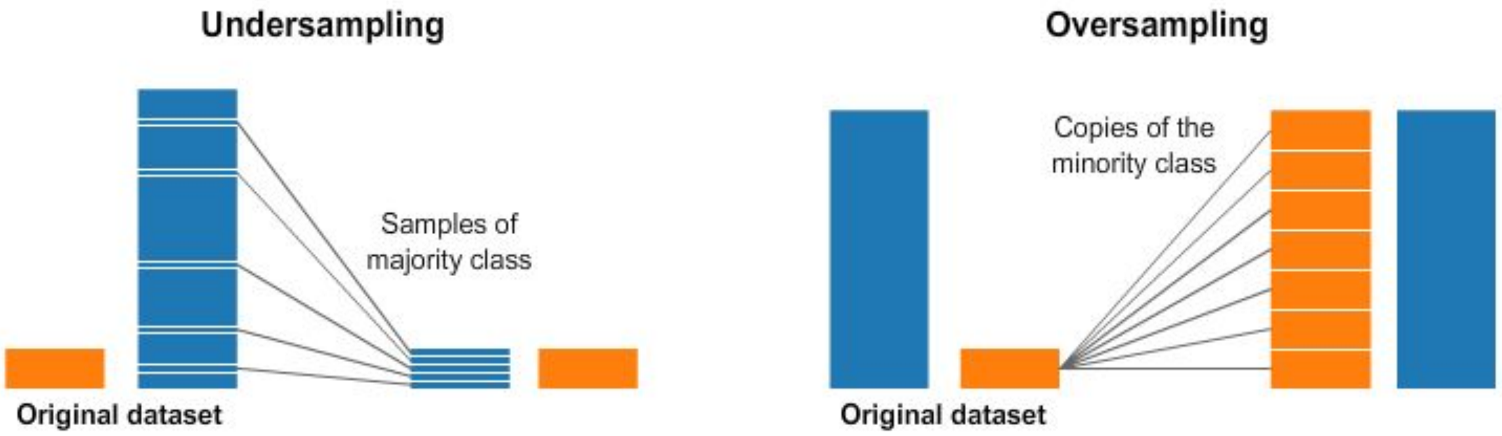
Undersampling	Oversampling
Remove samples from the majority class	Add more examples to the minority class



https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets#t1

2. Data-level methods: Resampling

Undersampling	**Oversampling**
Remove samples from the majority class	Add more examples to the minority class
Can cause loss of information	Can cause overfitting



https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets#t1

Undersampling: Tomek Links

- Find pairs of close samples of opposite classes
- Remove the sample of majority class in each pair
  - Pros: Make decision boundary more clear
  - Cons: Make model less robust

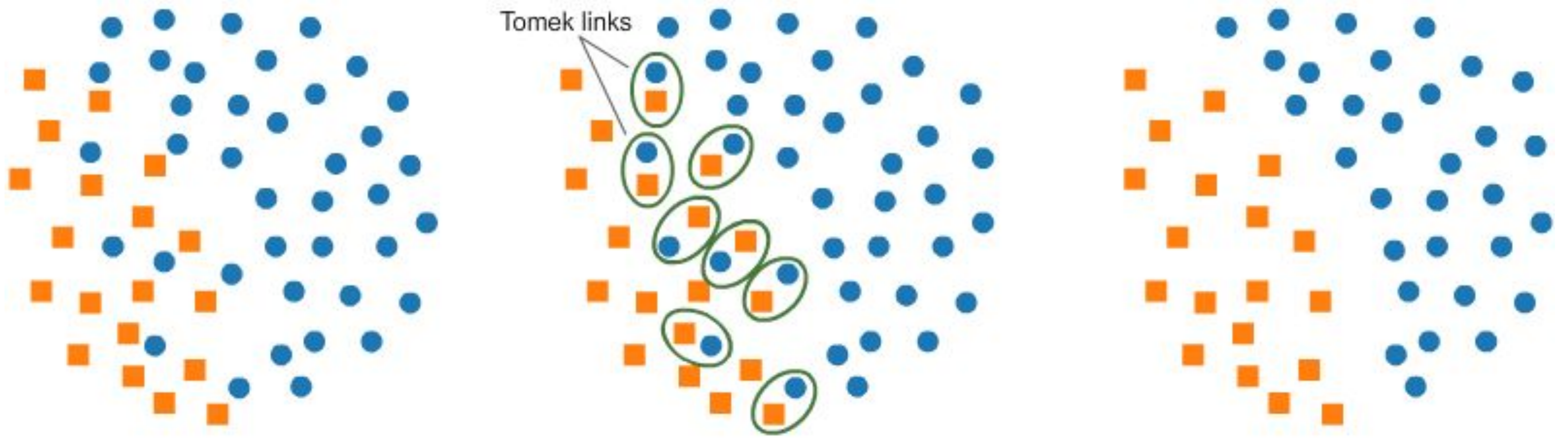


Image from https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets

Oversampling: SMOTE

- Synthesize samples of minority class as convex (~linear) combinations of existing points and their nearest neighbors of same class.

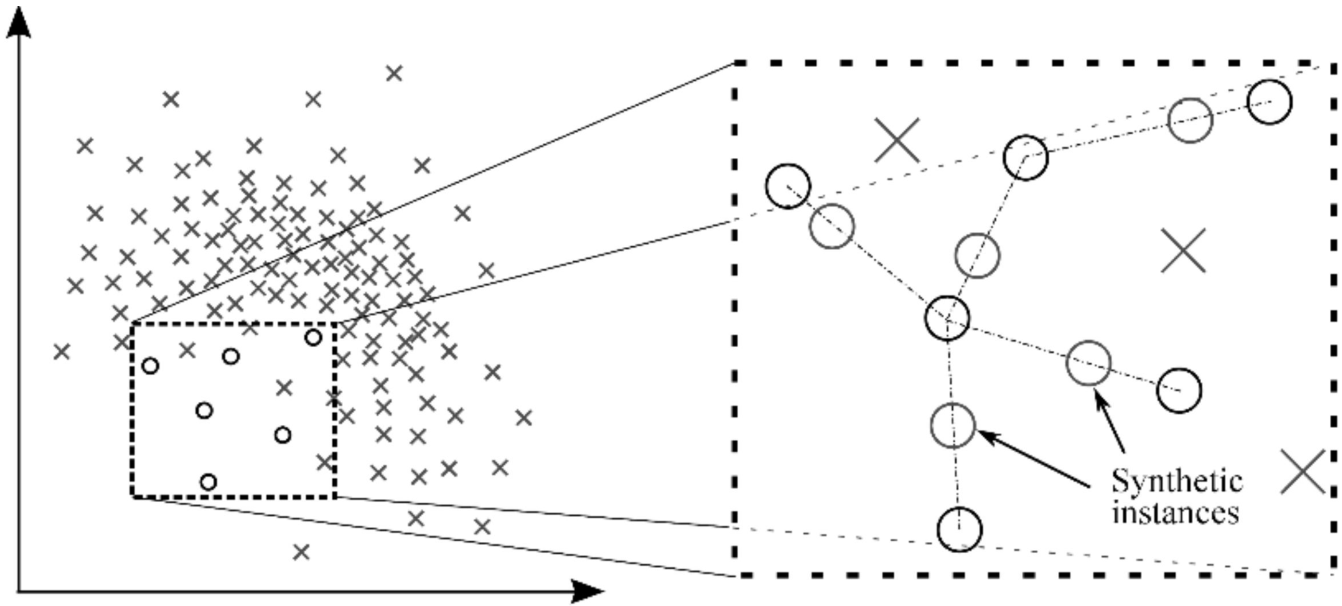


Image from [Analytics Vidhya]

Oversampling: SMOTE

Both SMOTE and Tomek links only work on low-dimensional data!

- Synthesize samples of minority class as convex (~linear) combinations of existing points and their nearest neighbors of same class.

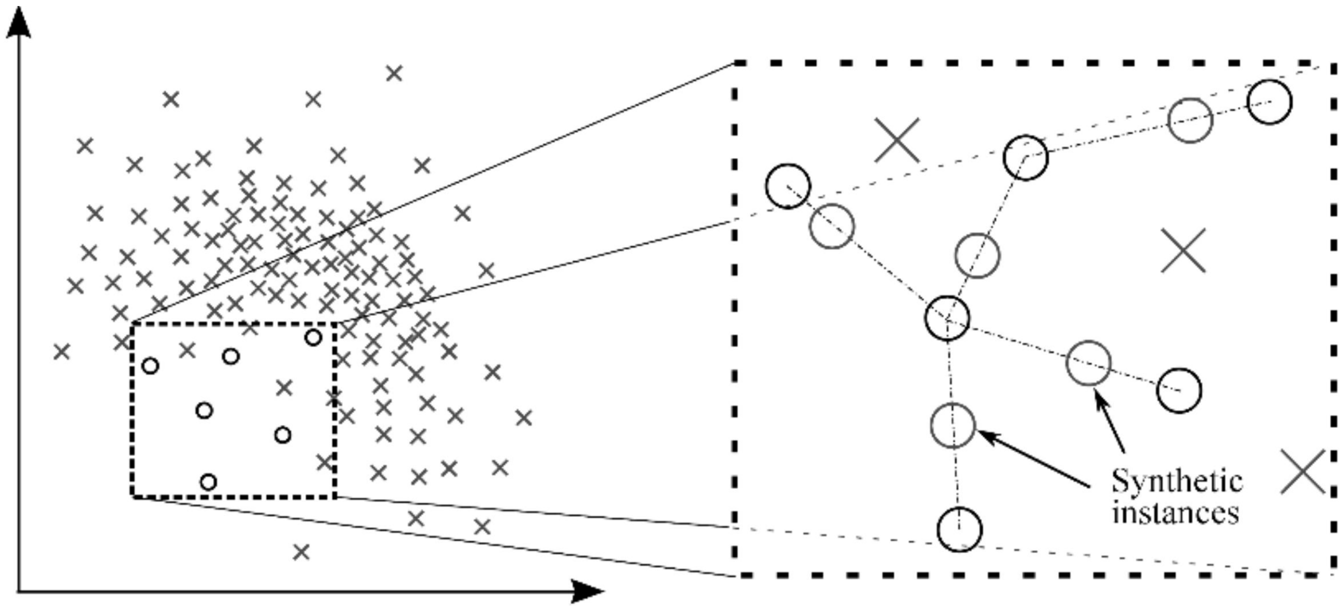


Image from [Analytics Vidhya]

3. Algorithm-level methods

- Naive loss: all samples contribute equally to the loss
- Idea: training samples we care about should contribute more to the loss

$$L(X;\ \theta) = \sum_{x} L(x\ ;\theta)$$

3. Algorithm-level methods

- Cost-sensitive learning
- Class-balanced loss
- Focal loss

Cost-sensitive learning

- C_{ij}: the cost if class i is classified as class j

	Actual NEGATIVE	Actual POSITIVE
Predicted NEGATIVE	$C(0,\ 0)\ =\ C_{00}$	$C(1,\ 0)\ =\ C_{10}$
Predicted POSITIVE	$C(0,\ 1)\ =\ C_{01}$	$C(1,\ 1)\ =\ C_{11}$

- The loss caused by instance x of class i will become the weighted average of all possible classifications of instance x.

$$L(x\ ;\theta)\ =\ \sum_{j} C_{ij} P(j\,|\,x;\ \theta)$$

Class-balance loss

- Give more weight to rare classes

Non-weighted loss

$$L(X;\ \theta) = \sum_i L(x_i; \theta)$$

Weighted loss

$$L(X;\ \theta) = \sum_i W_{y_i} L(x_i; \theta)$$

$$W_c = \frac{N}{number\ of\ samples\ of\ class\ C}$$

```
model.fit(features, labels, epochs=10, batch_size=32, class_weight={“fraud”: 0.9, “normal”:
0.1})
```

Focal loss

- Give more weight to difficult samples:
  - downweighs well-classified samples

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise,} \end{cases}$$

$$\mathrm{CE}(p_t) = -\log(p_t)$$

$$\mathrm{FL}(p_t) = -(1 - p_t)^{\gamma} \log(p_t)$$

Focal Loss for Dense Object Detection (Lin et al., 2017)

1. Data augmentation

"Data augmentation is the new feature engineering"

- Josh Wills, prev Director of Data Engineering @ Slack

Data augmentation: goals

- Improve model's performance overall or on certain classes
- Generalize better
- Enforce certain behaviors

Data augmentation

1. Simple label-preserving transformation
2. Perturbation
3. Data synthesis

Label-preserving: Computer Vision

Random cropping, flipping, erasing, etc.

Image from [An Efficient Multi-Scale Focusing Attention Network for Person Re-Identification]() (Huang et al., 2021)

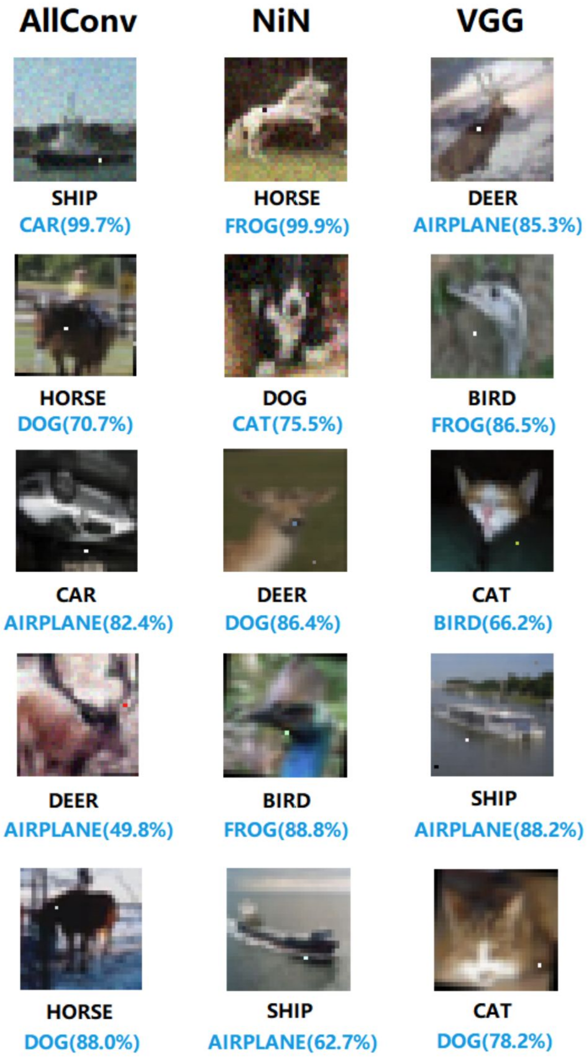
Label-preserving: NLP

Original sentences	I’m so happy to see you.
Generated sentences	I’m so **glad** to see you. I’m so happy to see **y’all**. I’m **very** happy to see you.

Perturbation: neural networks can be sensitive to noise

- 67.97% Kaggle CIFAR-10 test images
- 16.04% ImageNet test images

can be misclassified by changing just one pixel (Su et al., 2017)



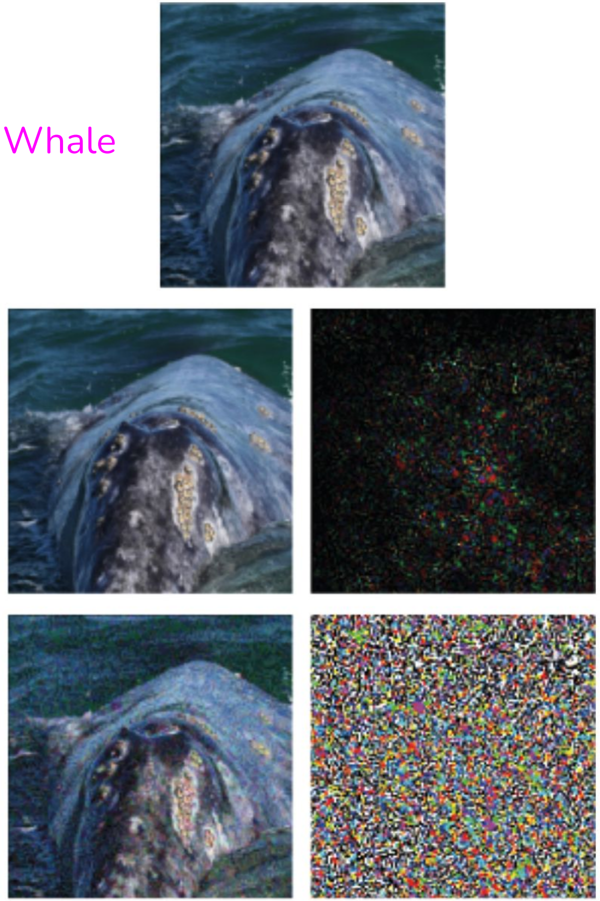
Perturbation: Computer Vision

- Random noise
- Search strategy
    - DeepFool (Moosavi-Dezfooli et al., 2016): find the minimal noise injection needed to cause a misclassification with high confidence.

Whale

Turtle
noise by DeepFool

Turtle
noise by fast gradient sign



Perturbation: NLP

- Random replacement
  - e.g. BERT (10% * 15% = 1.5%)

- 80% of the time: Replace the word with the `[MASK]` token, e.g., `my dog is hairy` → `my dog is [MASK]`
- 10% of the time: Replace the word with a random word, e.g., `my dog is hairy` → `my dog is apple`
- 10% of the time: Keep the word unchanged, e.g., `my dog is hairy` → `my dog is hairy`. The purpose of this is to bias the representation towards the actual observed word.

[BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#) (Devlin et al., 2018)

Data synthesis: NLP

- Template-based
  - Very common in conversational AI
- Language model-based

Template	Find me a [CUISINE] restaurant within [NUMBER] miles of [LOCATION].
Generated queries	• Find me a **Vietnamese** restaurant within **2** miles of **my office**. • Find me a **Thai** restaurant within **5** miles of **my home.** • Find me a **Mexican** restaurant within **3** miles of **Google headquarters**.

Data Synthesis: Computer Vision

- Mixup
  - Create convex combination of samples of different classes
    - Labels: cat [3], dog [4]
    - Mixup: 30% dog, 70% cat [0.3 * 3 + 0.7 * 4 = 3.7]

mixup: Beyond Empirical Risk Minimization (Zhang et al., 2017)

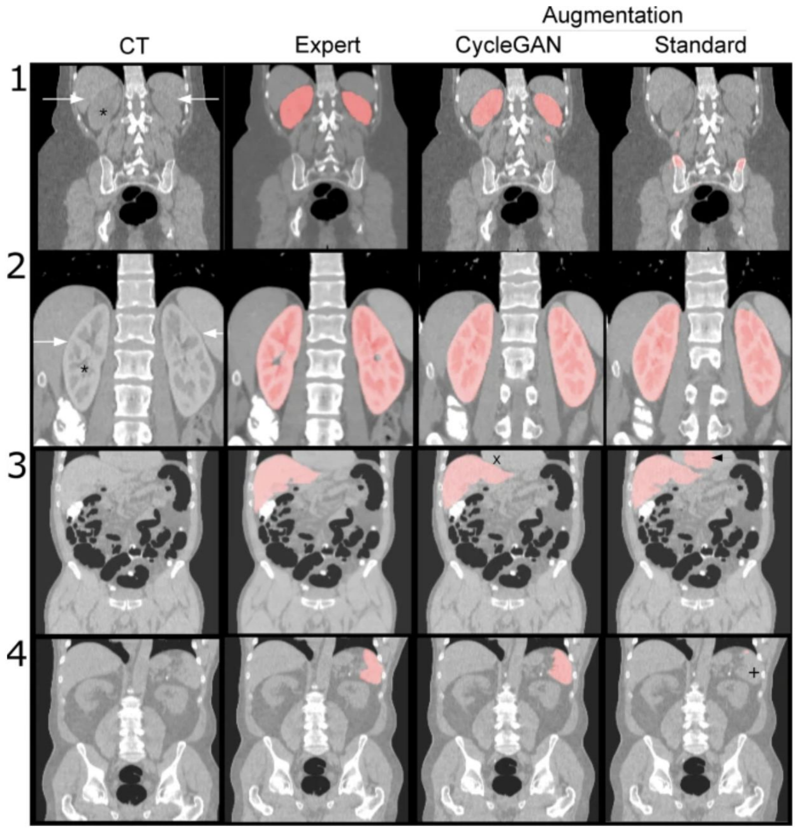
Data Synthesis: Computer Vision

- Mixup
  - Incentivize models to learn linear relationships
  - Improves generalization on speech and tabular data
  - Can be used to stabilize the training of GANs

https://forums.fast.ai/t/mixup-data-augmentation/22764

Data augmentation: GAN

Example: kidney segmentation with data augmentation by CycleGAN



Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks (Sandfort et al., 2019)

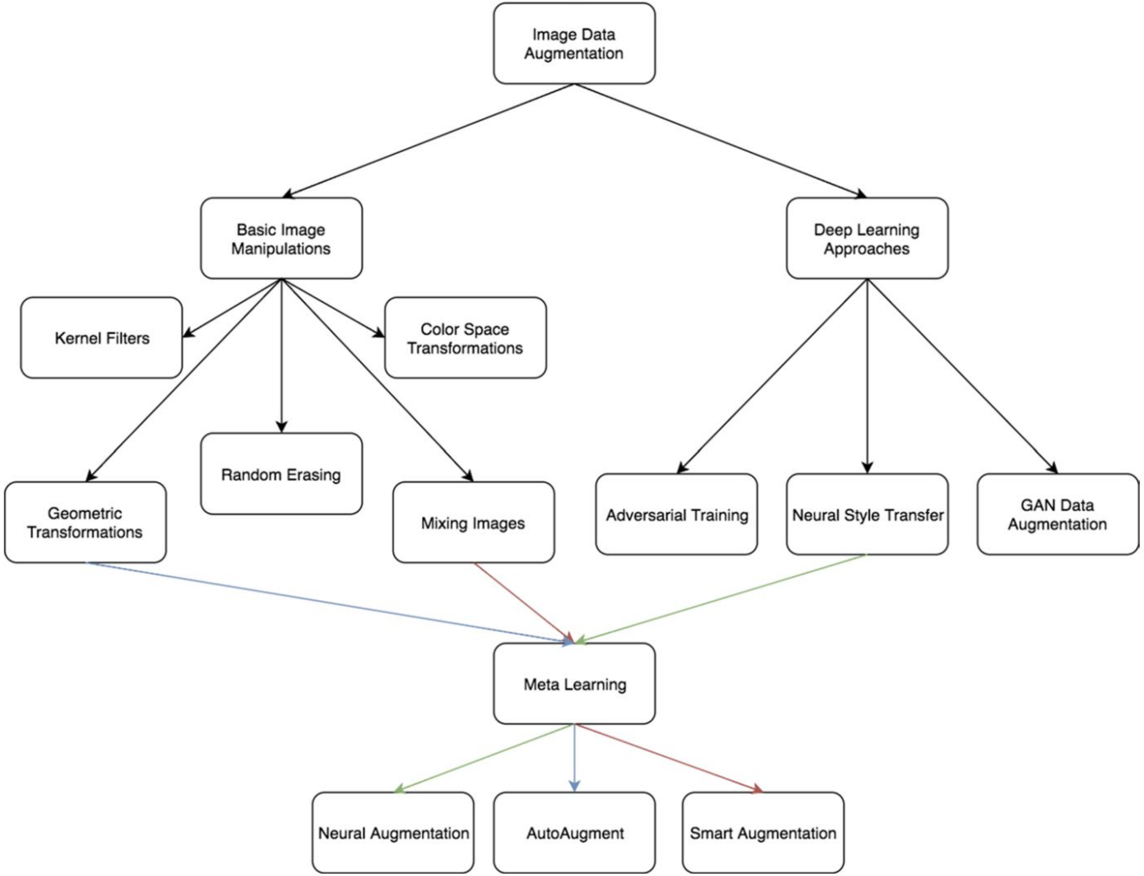


Fig. 2 A taxonomy of image data augmentations covered; the colored lines in the figure depict which data augmentation method the corresponding meta-learning scheme uses, for example, meta-learning using Neural Style Transfer is covered in neural augmentation [36]

A survey on Image Data Augmentation for Deep Learning (Connor Shorten & Taghi M. Khoshgoftaar, 2019)

Learned features
vs.
engineered features

Feature engineering

Wait, doesn't deep learning promise no more feature engineering?

Feature engineering

Wait, doesn't deep learning promise no more feature engineering?

- We're still very far from that point
- Many ML models in industry aren't deep learning

Engineered features: text

Stopword removal	I have a dog. He's sleeping.
Lemmatization	I have dog. He's sleeping.
Contraction	I have dog. He's sleep.
Punctuation	I have dog. He is sleep.
Lowercase	I have dog He is sleep
N-gram	i have dog he is sleep

features

samples

0	3	0	0	...
1	0	0	0	...
0	0	1	0	..
...	...	...	...	...

Classifier
(e.g. LogReg)

Features

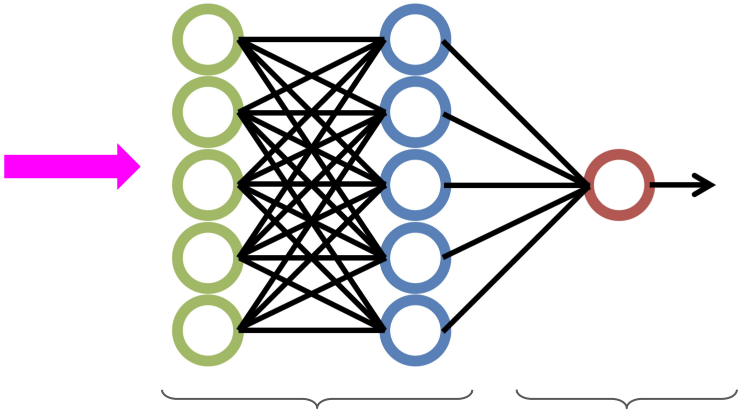
I	you	have	dog	cat	he	she	is	they	sleep	I, have	have, dog	good, dog	...
1	0	1	1	0	1	0	1	0	1	1	1	0	...

Learned features: text

one-hot encoding

samples

0	0	0	0	...
0	0	0	0	...
0	0	1	0	..
...	...	...	...	...



Text I have a dog. He’s sleeping.

One-hot

I	you	have	dog	cat	he	she	is	they	sleep	mom	food	yes	...
1	0	1	1	0	1	0	1	0	0	0	0	0	...

Learned features: spam classification

Comment ID	Time	User	Text	# ▲	# ▼	Link	# img	Thread ID	Reply to	# replies	...
93880839	2020-10-30 T 10:45 UTC	gitrekt	Your mom is a nice lady.	1	0	0	0	2332332	n0tab0t	1	...

User ID	Created	User	Subs	# ▲	# ▼	# replies	Karma	# threads	Verified email	Awards	...
4402903	2015-01-57 T 3:09 PST	gitrekt	[r/ml, r/memes, r/socialist]	15	90	28	304	776	No		...

Thread ID	Time	User	Text	# ▲	# ▼	Link	# img	# replies	# views	Awards	...
93883208	2020-10-30 T 2:45 PST	doge	Human is temporary, AGI is forever	120	50	1	0	32	2405	1	...

Feature engineering: spam classification

Even more features:

- Post frequency, max posts per day
- Post repetitiveness
- Language detection, typos, abnormal punctuations, ratio uppercase/lowercase
- IP, other users from the same IP
- NSFW words, blacklisted links
- Targeted users
- ...

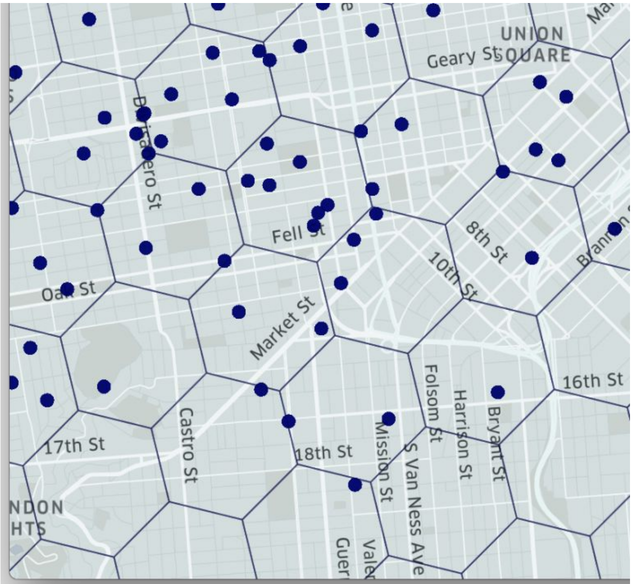
Feature engineering

- For complex tasks, number of features can go up to millions!
- Lots of ML production work involves coming up with new features
    - Fraudsters come up with new techniques very fast, so need to come up with new features very fast to counter
- Often require subject matter expertise

Breakout exercise

Group of 4, 10 minutes

- Imagine you're building a model to predict the trip duration given a pickup and a drop-off location. What features would you use?
- Things to consider:
  - The distance between 2 points is not the same the distance between 2 locations on the map
  - Uber divides their maps into many hexagons
  - Events/environments that can affect a trip duration
  - What features you can pull from a database?
  - What features would you need to compute in (near) real-time?



[Inspired by [Kaggle's taxi trip duration dataset]]

Image from https://eng.uber.com/h3/

Feature engineering operations

Kinbert Chou

Common feature engineering ops

1. Handling missing values
2. Scaling
3. Discretization
4. Categorical features
5. Feature crossing
6. Positional embeddings

Handling missing values

- Not all missing values are equal
  - Missing not at random (MNAR)
  - Missing at random (MAR)
  - Missing completely at random (MCAR)

Handling missing values

Missing not at random – when a value is missing due to the value itself

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B	($350,000?)		2	Engineer	Yes
5	35	B	($350,000?)	Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Handling missing values

Missing at random – when a value is missing due to another observed variable

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Handling missing values

Missing completely at random – there is no pattern to which values are missing

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Handling missing values

- Deletion – removing data with missing entries
- Imputation – filling missing fields with certain values

Many people prefer deletion not because it's better, but it's easier to do

Handling missing values

- Deletion
  - Column deletion – remove columns with too many missing entries
    - drawbacks – even if half the values are missing, the remaining data still potentially useful information for predictions
    - e.g. even if over half the column for ‘Marital status’ is missing, marital status is still highly correlated with house purchasing
  - Row deletion

Marital status
Married
Single
Single

Handling missing values

- Deletion
  - Column deletion
  - Row deletion

Handling missing values

- Row deletion
  - Good for: data missing completely at random (MCAR) and few values missing

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1	39	A	150,000	Married	1	Engineer	No
2	27	B	50,000	Single	0	Teacher	No
3		A	100,000	Married	2		Yes
4	40	B	75,000	Married	2	Engineer	Yes
5	35	B	35,000	Single	0	Doctor	Yes
6	32	A	50,000	Married	0	Teacher	No
7	33	B	60,000	Single	2	Teacher	No
8	20	B	10,000	Single	1	Student	No

Handling missing values

- Row deletion
  - Bad when many examples have missing fields

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Handling missing values

- Row deletion
    - Bad for: missing values are not at random (MNAR)
    - Missing information is information itself

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B	($350,000?)		2	Engineer	Yes
5	35	B	($350,000?)	Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Handling missing values

- Row deletion
  - Bad for: missing data at random (MAR)
  - Can potentially bias data – we've accidentally removed all examples with gender 'A'

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Imputation

- Fill missing fields with certain values
  - Defaults
    - E.g. 0, or the empty string, etc.
  - Statistical measures – mean, median, mode
    - e.g. if a day in July is missing its temperature value, fill it with the median temperature in July

Imputation

- Fill missing fields with certain values
  - Defaults
    - E.g. 0, or the empty string, etc.
  - Statistical measures – mean, median, mode
    - e.g. if a day in July is missing its temperature value, fill it with the median temperature in July

Avoid filling missing values with possible values!

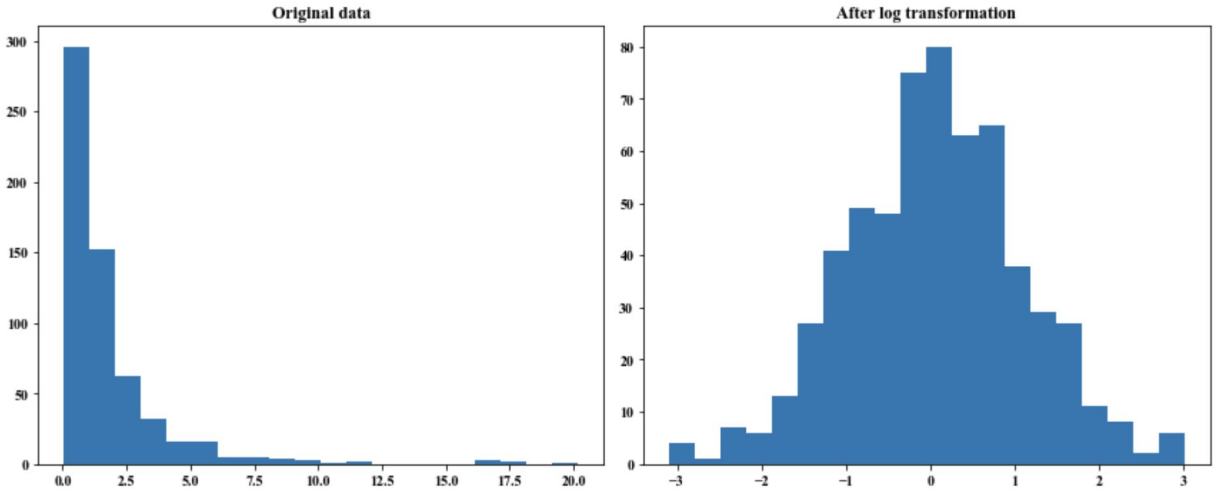
Scaling

Types of scaling

scaling type	use case
min/max normalization	Any -- no assumptions about variables
z-score normalization	When variables follow a normal distribution
log scaling	When variables follow an exponential distribution

Log scaling

- Help with skewed data
- Often gives performance gain



Scaling

scaling type	use case
min/max normalization	Any -- no assumptions about variables
z-score normalization	When variables follow a normal distribution
log scaling	When variables follow an exponential distribution

- scaling can be a common source of data leakage

Scaling

scaling type	use case
min/max normalization	Any -- no assumptions about variables
z-score normalization	When variables follow a normal distribution
log scaling	When variables follow an exponential distribution

- scaling can be a common source of data leakage
- scaling variables requires global statistics

Discretization

- Turning a continuous feature into a discrete feature (quantization)

Discretization

- Turning a continuous feature into a discrete feature (quantization)
- Create buckets for different ranges
  - Incorporate knowledge/expertise about each variable by constructing specific buckets

Discretization

- Turning a continuous feature into a discrete feature (quantization)
- Create buckets for different ranges
  - Incorporate knowledge/expertise about each variable by constructing specific buckets
- Examples
  - Income
    - Lower income: $x < \$35{,}000$
    - Middle income: $\$35{,}000 < x < \$100{,}000$
    - High income: $x > \$100{,}000$

Discretization

- Turning a continuous feature into a discrete feature (quantization)
- Create buckets for different ranges
    - Incorporate knowledge/expertise about each variable by constructing specific buckets
- Examples
    - Income
        - Lower income: x < $35,000
        - Middle income: $35,000 <= x < $100,000
        - High income: x >= $100,000
    - Age
        - Minors: x < 18
        - College: 18 <= x < 22
        - Young adult: 22 <= x < 30
        - 30 <= x < 40
        - 40 <= x < 65
        - Seniors: x >= 65

Encoding Categorical Features

- Example: you want to build a recommendation system for Amazon
  - There are over 2 million brands that we need to recommend

Encoding Categorical Features

How do we encode the different brands/vendors?

Encoding Categorical Features

- one-hot encoding!

How do we encode the different brands/vendors?

Encoding Categorical Features

- one-hot encoding!

How do we handle a new brand that wants to join Amazon?

Encoding Categorical Features

- one-hot encoding!
- encode unseen brands with “UNKNOWN”

How do we handle a new brand that wants to join Amazon?

Encoding Categorical Features

- one-hot encoding!
- encode unseen brands with “UNKNOWN”

Problem! “UNKNOWN” was not seen during training, so none of the products in this category are being recommended

Encoding Categorical Features

- one-hot encoding!
- encode unseen brands with “UNKNOWN”

Fix – encode brands as themselves, group bottom-performing 1% of brands as “UNKNOWN”

Encoding Categorical Features

- one-hot encoding!
- encode unseen brands with “UNKNOWN”
- Group bottom 1% of brands and newcomers into “UNKNOWN” category

Encoding Categorical Features

- one-hot encoding!
- encode unseen brands with “UNKNOWN”
- Group bottom 1% of brands and newcomers into “UNKNOWN” category

Alert! Nike wants to join Amazon as a new vendor

Encoding Categorical Features

- one-hot encoding!
- encode unseen brands with “UNKNOWN”
- Group bottom 1% of brands and newcomers into “UNKNOWN” category
- Problem – this treats all newcomers the same as unpopular brands on the platform

Encoding Categorical Features

How do we implement a flexible method of handling new brands as they are introduced to our system?

Encoding New Categories

1. Represent each category with its attribute
    a. E.g. to represent a brand, use features: yearly revenue, company size, etc..
2. Hashing trick

Encoding Categorical Features

- Hashing – use a hash function to hash categories to different indexes

Encoding Categorical Features

- Hashing – use a hash function to hash categories to different indexes
  - e.g. hash(“Nike”) = 0, hash(“Adidas”) = 27, etc…

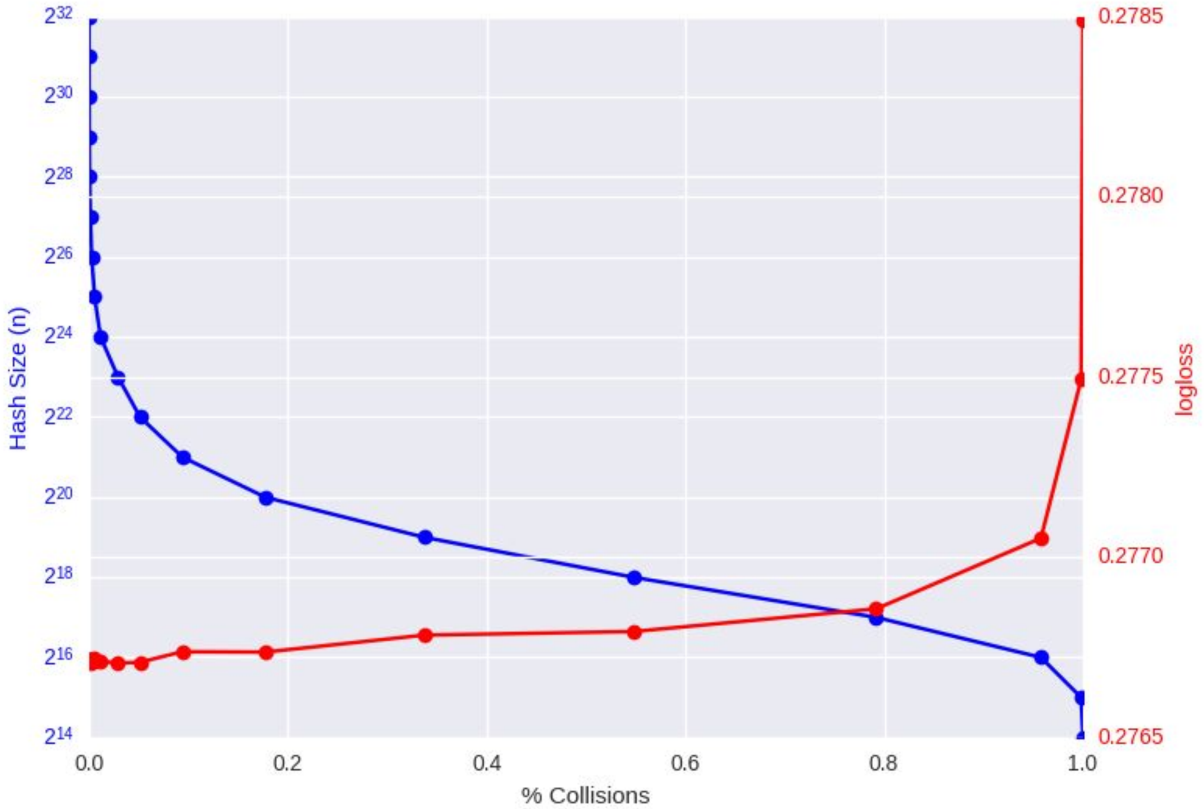
Encoding Categorical Features

- Hashing – use a hash function to hash categories to different indexes
    - e.g. hash(“Nike”) = 0, hash(“Adidas”) = 27, etc…
- Benefits – you can choose how large the hash space is

Encoding Categorical Features

- Hashing – use a hash function to hash categories to different indexes
  - e.g. hash(“Nike”) = 0, hash(“Adidas”) = 27, etc…
- Benefits – you can choose how large the hash space is
- Drawbacks – two categories being hashed to the same index

Encoding Categorical Features



Encoding Categorical Features

- Choose a hash space large enough to reduce collisions
- Choose functions with properties beneficial to your use case
  - Locality-sensitive hashing

Hashing Trick Takeaways

- Hashing trick considered “hacky” by academics
- Widely used in industry and in machine learning frameworks
- Useful in practice for continual learning in production

Feature Crossing

- Combine two or more features to create a new feature

Marriage	Single	Married	Single	Single	Married
Children	0	2	1	0	1
Marriage & children	Single, 0	Married, 2	Single, 1	Single, 0	Married, 1

Feature Crossing

- Helps models learn non-linear relationships between variables
- Warning – feature crossing can blow up your feature space
  - e.g. Feature A and B both have 100 categories → Feature A x B will have 10,000 categories
  - Need even more data to learn this new feature space
  - Blowing up feature space can increase risk of overfitting

Very common in RecSys & CTR with models like DeepFM and xDeepFM

Positional Embeddings

- Popularized in [Attention is All You Need](#) paper
- Similar to word embeddings
  - Can be either learned or fixed

Word embedding matrix

emb for word “food” (index 0)

word embedding size

0.001	0.023	-0.003	0.004	...
0...	0...	-0...	0...	...
0..	-0...	-1	-0...	..
...	...	...	...	...

Vocab

food	i	you	are	....
0	1	2	3	....

Position embedding matrix

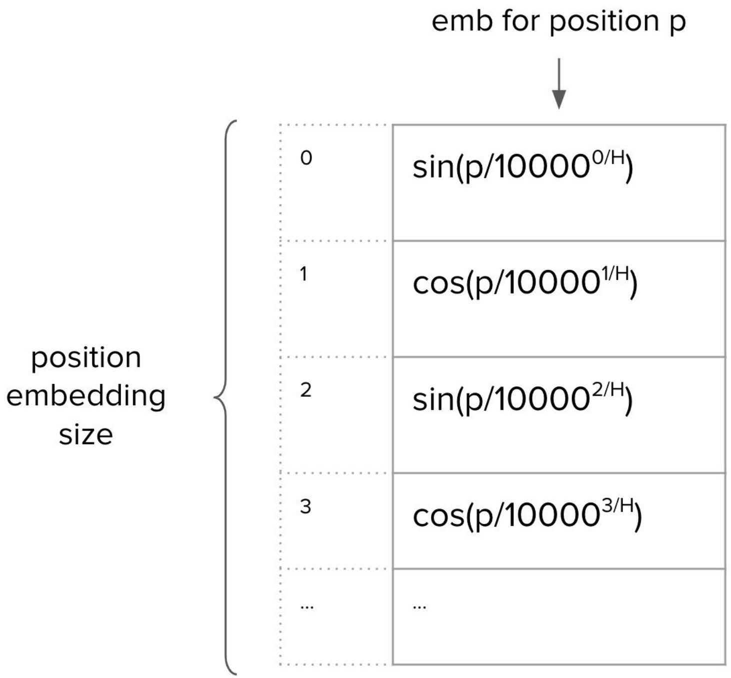
emb for position 0

position embedding size

0.001	0.0023	-0.003	0.004	...
0...	0...	-0...	0...	...
0..	-0...	-1	-0...	..
...	...	...	...	...

Positional Embeddings

- Fourier features



Positional Embeddings

Why do we need position embeddings?

Positional Embeddings

Why do we need position embeddings?

- Traditional architectures (RNNs, LSTMs) process tokens sequentially
- Transformers process tokens in parallel → need to communicate sequential nature of human language to model

Data leakage

Data leakage

- Some form of the label “leaks” into the features
- This same information is not available during inference

Data leakage: example 1

- Problem: detect lung cancer from CT scans
- Data: collected from hospital A
- Performs well on test data from hospital A
- Performs poorly on test data from hospital B

Patient ID	Date	Doctor note	Medical record	Scanner type	CT scan

Data leakage: example 1

- Problem: detect lung cancer from CT scans
- Data: collected from hospital A
- Performs well on test data from hospital A
- Performs poorly on test data from hospital B

Patient ID	Date	Doctor note	Medical record	Scanner type	CT scan

At hospital A, when doctors suspect that a patient has lung cancer, they send that patient to a higher-quality scanner

Data leakage: example 2

- Problem: predicting how many views an article will get
- Data: historical data on the site
- Where might data leakage come from?

Article ID	Date	Title	Article	Author	Language	Translations

Data leakage: example 2

- Problem: predicting how many views an article will get
- Data: historical data on the site

Not leakage because author popularity also available during inference

Article ID	Date	Title	Article	Author	Language	Translations

The site only translate articles that are already gaining attention

Data leakage: Kaggle edition

What will be done about data leaks?
Louka Ewington-Pitsos · Last comment 2Y ago by Gunes Evitan
23 · 7 comments

Time for Kaggle to clarify
bluetrain · Last comment 2Y ago by Vishal
56 · 8 comments

There is something new about Leaks ?
Haythem Tellili · Last comment 2Y ago by Oskin Nikita
4 · 1 comment

New leakage web finded
Zhang Yunfei · Last comment 2Y ago by Zhang Yunfei
19 · 19 comments

[ASHRAE - Great Energy Predictor III]

Zidmie
Topic Author
3rd place

The leak explained!
60
Posted in liverpool-ion-switching 2 years ago

The cat5 data (category with 10 open channels) is very similar to the addition of two signals of cat4 data, as several teams noticed. But also, we can find that the data from 4000001 to 4100000 (cat4) was used to create the data from 5700001 to 5800000 : openchannels(5700001:5800000) = openchannels(4000001 to 4100000) + openchannels of other cat4 data (that I didn't really look for). So we can just substract data openchannels(4000001 to 4100000) from the private LB data. Then we have cat4 data, much easier to predict. By using this method, we just reached a score of 0.9543. I guess by digging further, it can lead to the score of 0.985!

[University of Liverpool - Ion Switching]

Exercises

1. What are the causes of data leakage?
2. How to detect data leakage?

Machine Learning Systems Design

Next class: Model development

No class next Monday!

Stanford University