

01418262 Machine Learning Systems

Feature Engineering

Sarach Tuomchomtam, Ph.D.

sarach.t@ku.th

Feature Engineering

Having the right features tends to give them the biggest performance boost compared to clever algorithmic techniques.

State-of-the-art model architectures can still perform poorly if they don't use a good set of features.

Practical Lessons from Predicting Clicks on Ads at Facebook

Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu*, Tao Xu*, Yanxin Shi*,
Antoine Atallah*, Ralf Herbrich*, Stuart Bowers, Joaquin Quiñero Candela
Facebook
1601 Willow Road, Menlo Park, CA, United States
{panjunfeng, oujin, joaquin, sbowers}@fb.com

ABSTRACT

Online advertising allows advertisers to only bid and pay for measurable user responses, such as clicks on ads. As a consequence, click prediction systems are central to most online advertising systems. With over 750 million daily active users and over 1 million active advertisers, predicting clicks on Facebook ads is a challenging machine learning task. In this paper we introduce a model which combines decision trees with logistic regression, outperforming either of these methods on its own by over 3%, an improvement with significant impact to the overall system performance. We then explore how a number of fundamental parameters impact the final prediction performance of our system. Not surprisingly, the most important thing is to have the right features: those capturing historical information about the user or ad dominate other types of features. Once we have the right features and the right model (decisions trees plus logistic regression), other factors play small roles (though even small improvements are important at scale). Picking the optimal handling for data freshness, learning rate schema and data sampling improve the model slightly, though much less than adding a high-value feature, or picking the right model to begin with.

efficiency of the marketplace.

The 2007 seminal papers by Varian [11] and by Edelman et al. [4] describe the bid and pay per click auctions pioneered by Google and Yahoo! That same year Microsoft was also building a sponsored search marketplace based on the same auction model [9]. The efficiency of an ads auction depends on the accuracy and calibration of click prediction. The click prediction system needs to be robust and adaptive, and capable of learning from massive volumes of data. The goal of this paper is to share insights derived from experiments performed with these requirements in mind and executed against real world data.

In sponsored search advertising, the user query is used to retrieve candidate ads, which explicitly or implicitly are matched to the query. At Facebook, ads are not associated with a query, but instead specify demographic and interest targeting. As a consequence of this, the volume of ads that are eligible to be displayed when a user visits Facebook can be larger than for sponsored search.

In order to tackle a very large number of candidate ads per request, where a request for ads is triggered whenever a user

Feature Engineering

Outline

Learned vs. Engineered Features

Common Operations

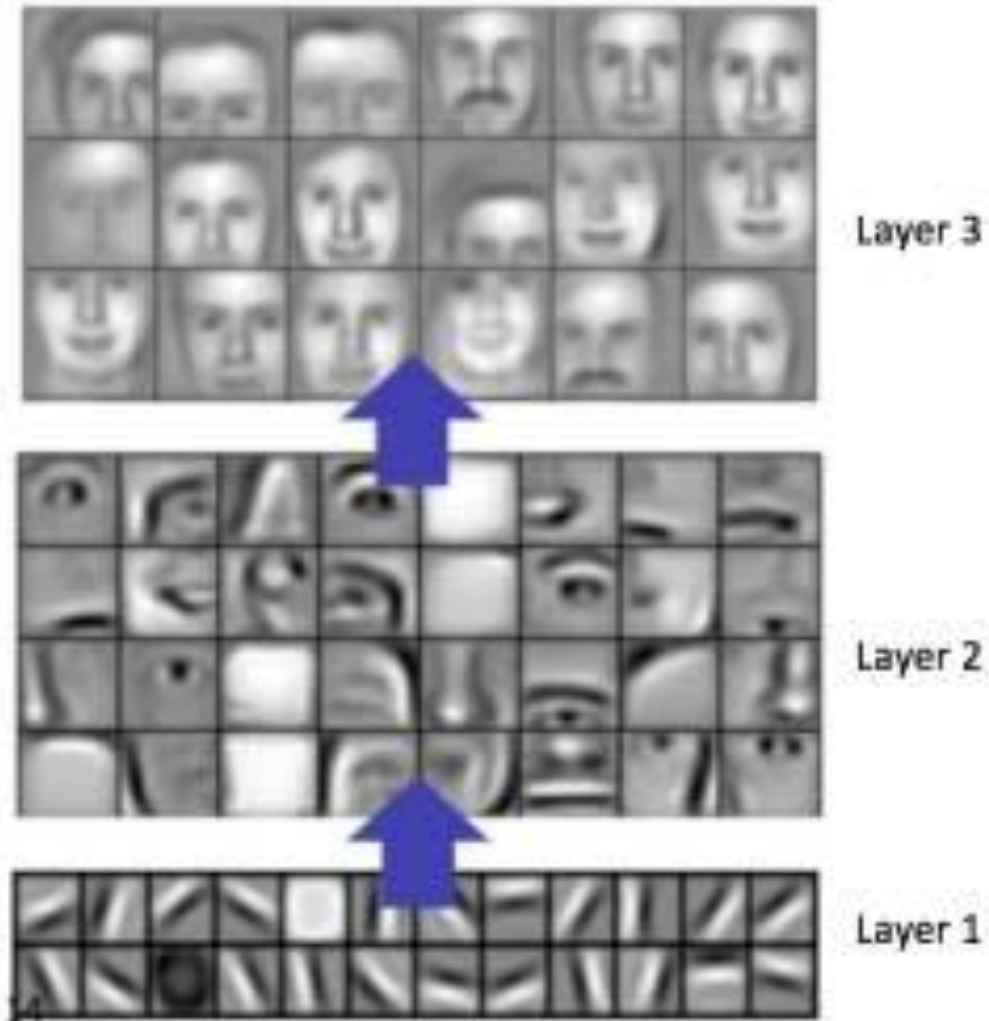
Data Leakage

Engineering Good Features

Learned vs. Engineered Features

“Why do we have to worry about feature engineering? Doesn't deep learning promise us that we no longer have to engineer features?”

Learned vs. Engineered Features



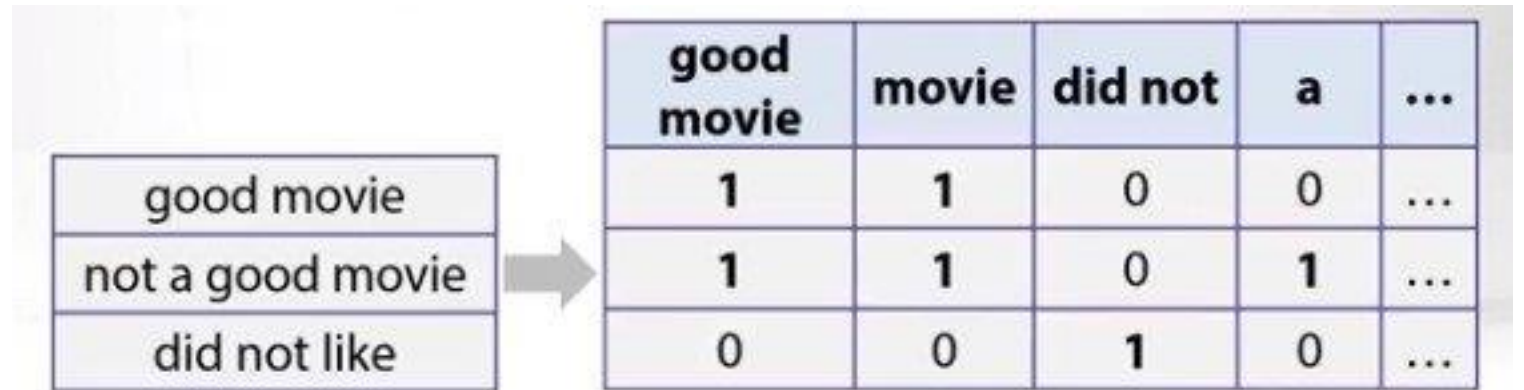
Feature Engineering

N-gram Vectorization

A contiguous sequence of n items from a given sample of text.

Create a vocabulary that maps each n -gram to an index.

Convert each post into a vector based on its n -grams' indices.







The diagram illustrates the process of N-gram vectorization. On the left, a table lists three text samples. An arrow points from this table to a larger matrix on the right. The matrix's columns represent the N-grams 'good movie', 'movie', 'did not', 'a', and '...', which are defined in the top row. The rows represent the three text samples, with binary values (0 or 1) indicating the presence of each N-gram. For example, the first sample 'good movie' has a 1 in the 'good movie' and 'movie' columns, and 0s elsewhere. The second sample 'not a good movie' has 1s in the 'good movie' and 'movie' columns, and a 1 in the 'a' column. The third sample 'did not like' has a 1 in the 'did not' column and 0s elsewhere.



good movie	1	1	0	0	...
not a good movie	1	1	0	1	...
did not like	0	0	1	0	...

Domain-specific Features

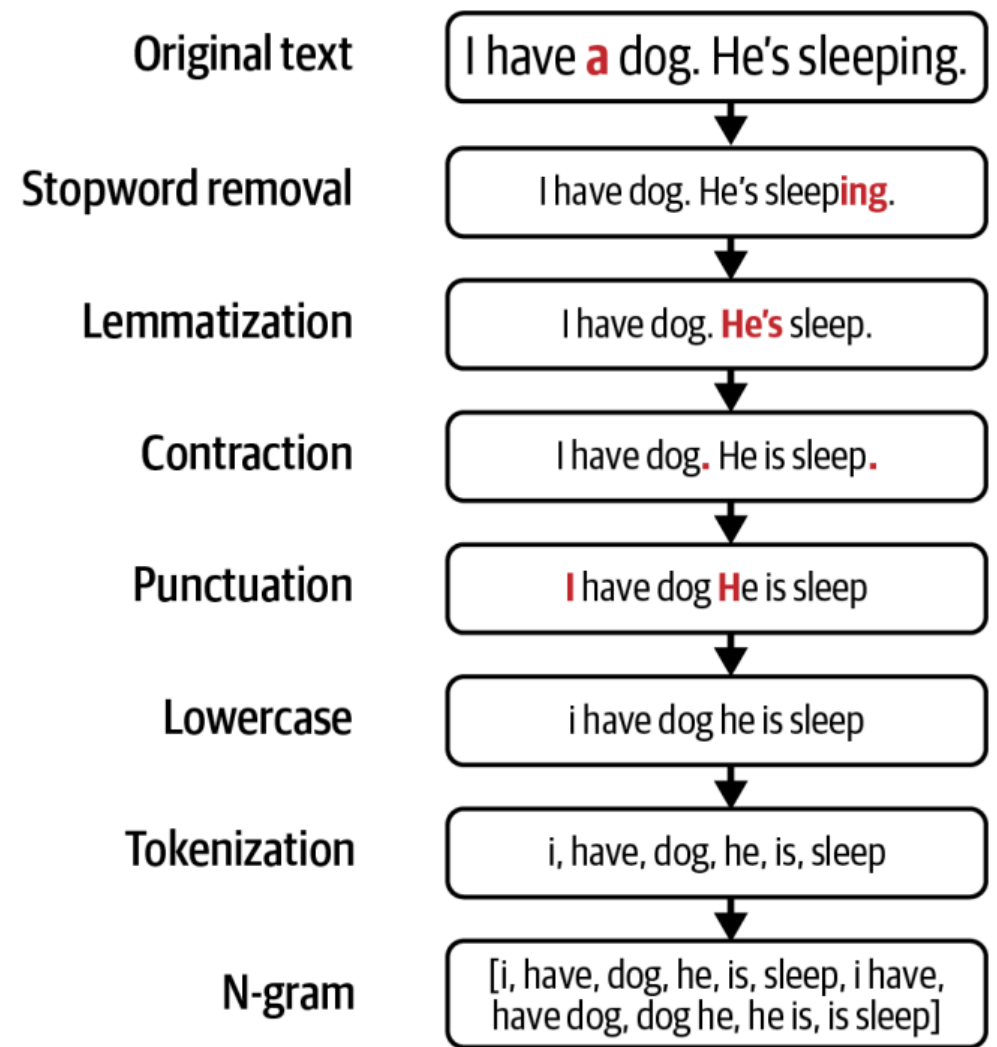
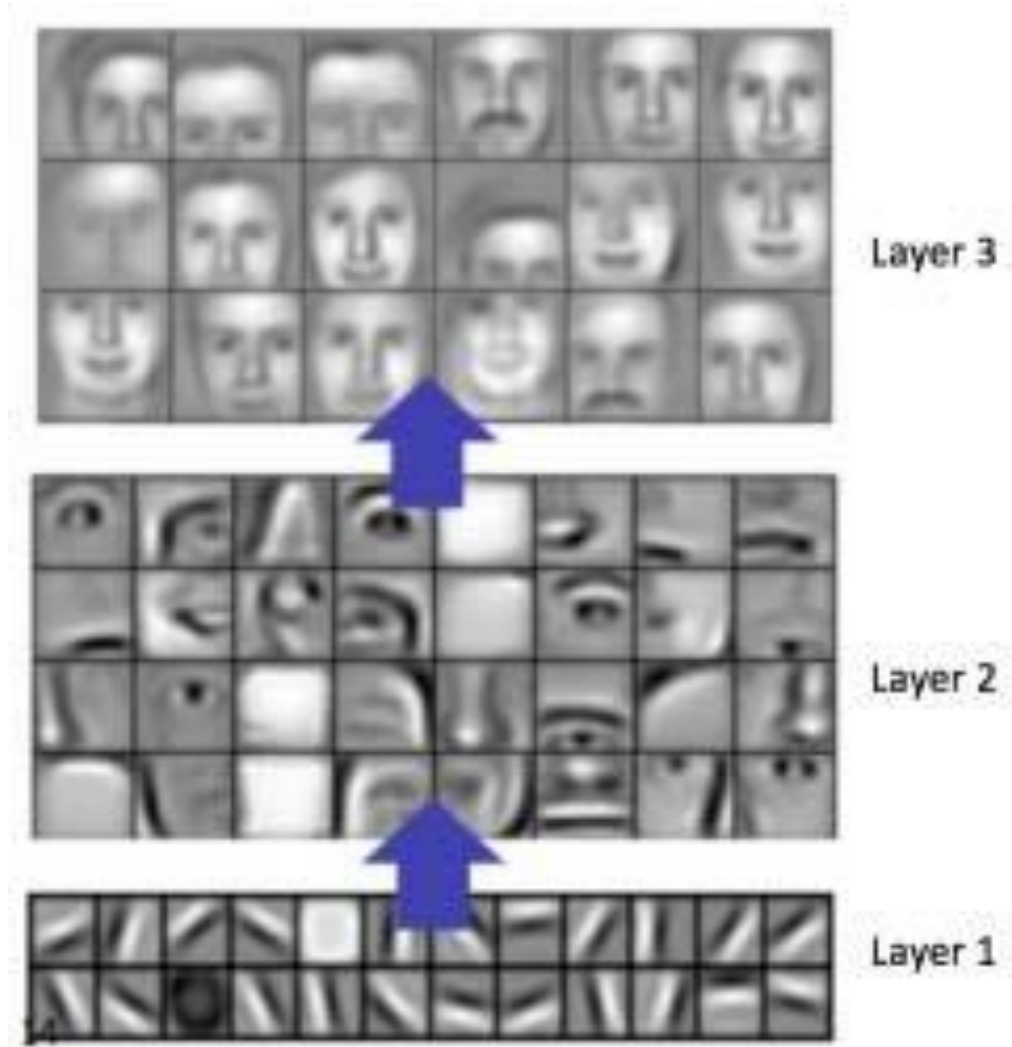
When detecting whether a comment is spam or not, on top of the text in the comment itself, you might want to use other information about:

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

Learned vs. Engineered Features



Common Operations

There have been many techniques developed to streamline the process.

This list is nowhere near being comprehensive, but it does comprise some of the most common and useful operations.

- 1. Handling missing values**
- 2. Scaling**
- 3. Discretization**
- 4. Encoding categorical features**
- 5. Feature crossing**

Common Operations

Handling Missing Values

Not all types of missing values are equal.

Consider the task of predicting whether someone is going to buy a house in the next 12 months.

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

Types of Missing Values

The names for these types are a little bit confusing.

When encountering missing values, you can either fill in the missing values with certain values (imputation) or remove the missing values (deletion).

- **Missing not at random (MNAR)**
“Income of whom failed to report tends to be higher than that of those who did disclose.”
- **Missing at random (MAR)**
“People of gender A in this survey don’t like disclosing their age.”
- **Missing completely at random (MCAR)**
“People just forget to fill in that value sometimes for no particular reason.”

Handling Missing Values

Imputation

Deleting data can lead to losing important information and introduce biases.

Imputation means “fill them with certain values.”

Risk injecting your own bias into and adding noise to your data.

- **Fill in with their defaults**

Such as empty string, mean, median, or mode.

- **Avoid filling with possible values**

Such as filling the missing number of children with 0.

Deletion

Column deletion

- If a variable has too many missing values, just remove that variable.
- Might remove important information and reduce the accuracy.

Row deletion

- If a sample has missing value(s), just remove that sample.
- Can remove important information and create biases in your model.

Common Operations

Scaling

ML model won't understand that 150,000 and 40 represent different things.

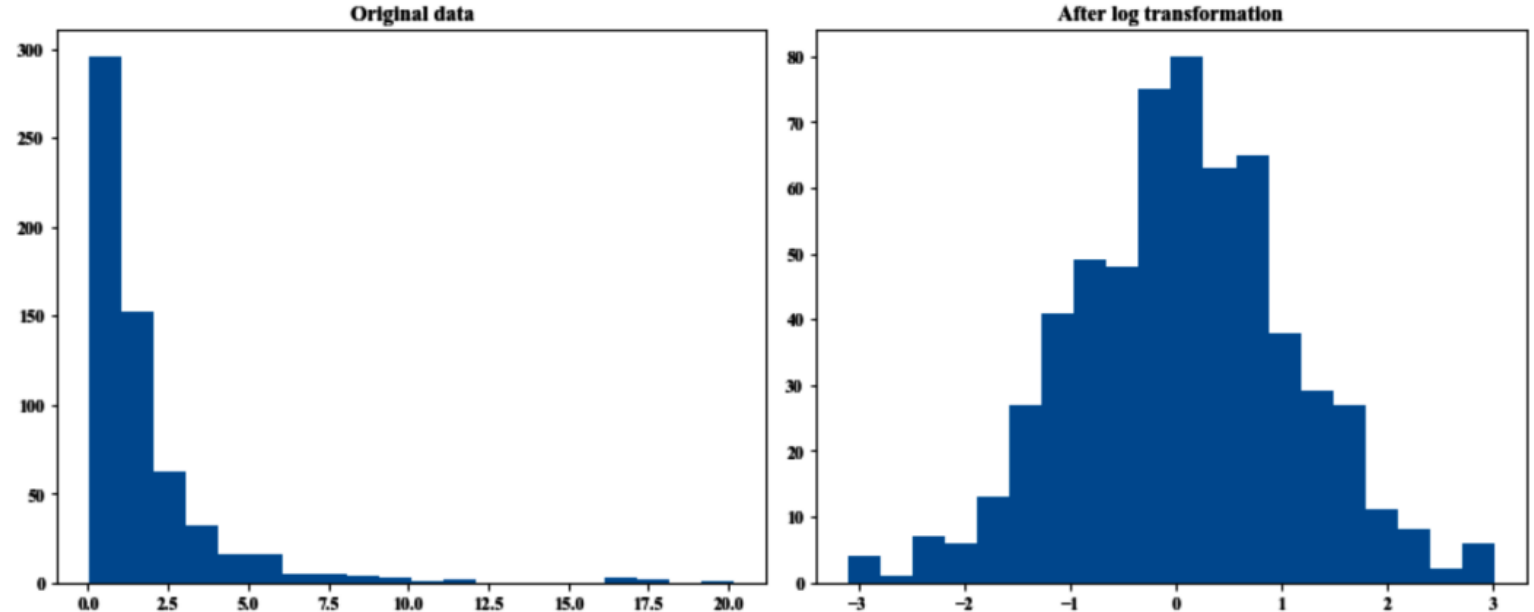
It's important to scale them to be similar ranges.

To help mitigate the skewness, a technique commonly used is log transformation.

0 to 1:
$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

a to b:
$$x' = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)}$$

Zero-mean variance:
$$x' = \frac{x - \bar{x}}{\sigma}$$



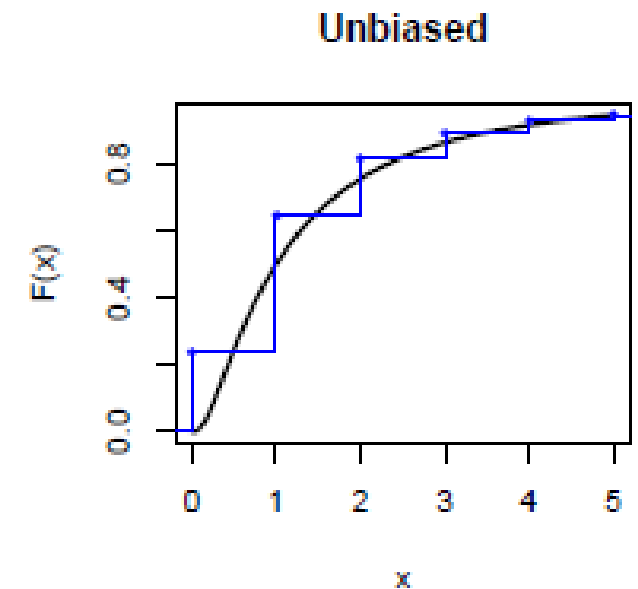
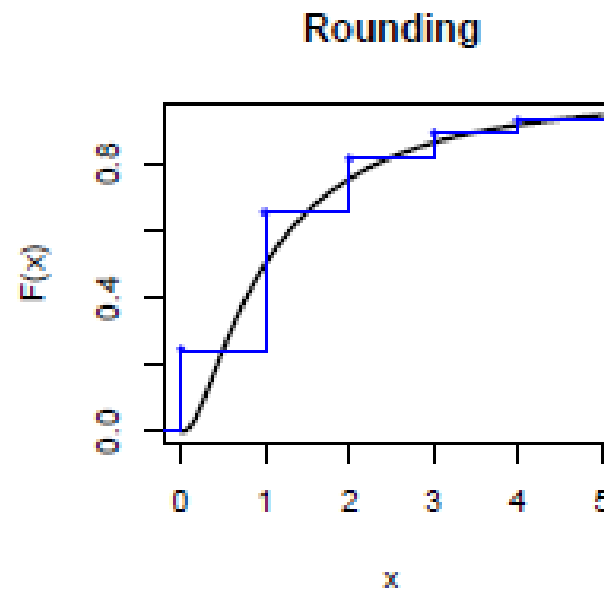
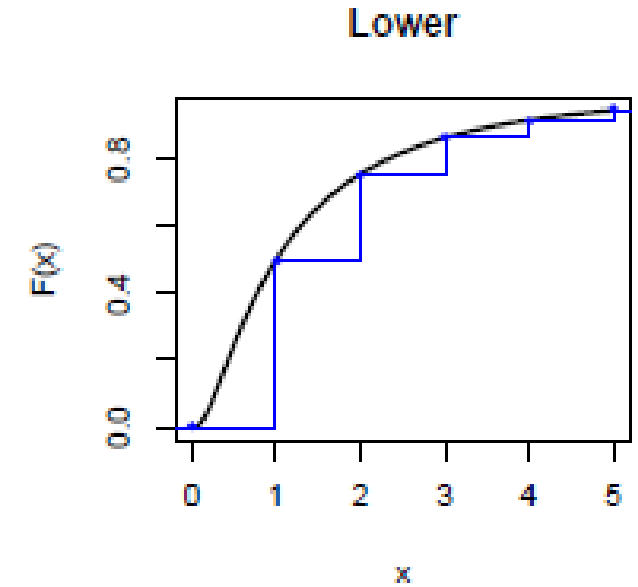
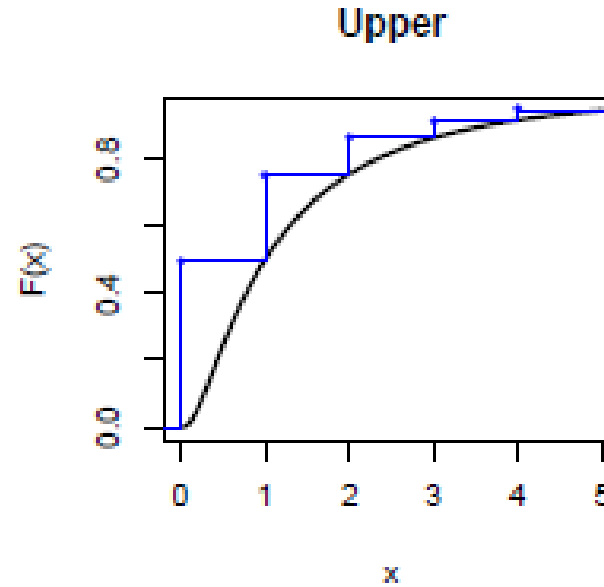
Common Operations

Discretization

The process of turning a continuous feature into a discrete feature, also known as quantization or binning.

This is done by creating buckets for the given values.

Introduces discontinuities at the category boundaries.



Common Operations

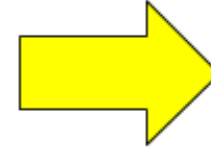
Encoding Categorical Features

People tend to assume that categories are static, they are not.

For some problems, new categories are being created all the time.

One solution to this problem is the hashing trick.

Color
Red
Red
Yellow
Green
Yellow



Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

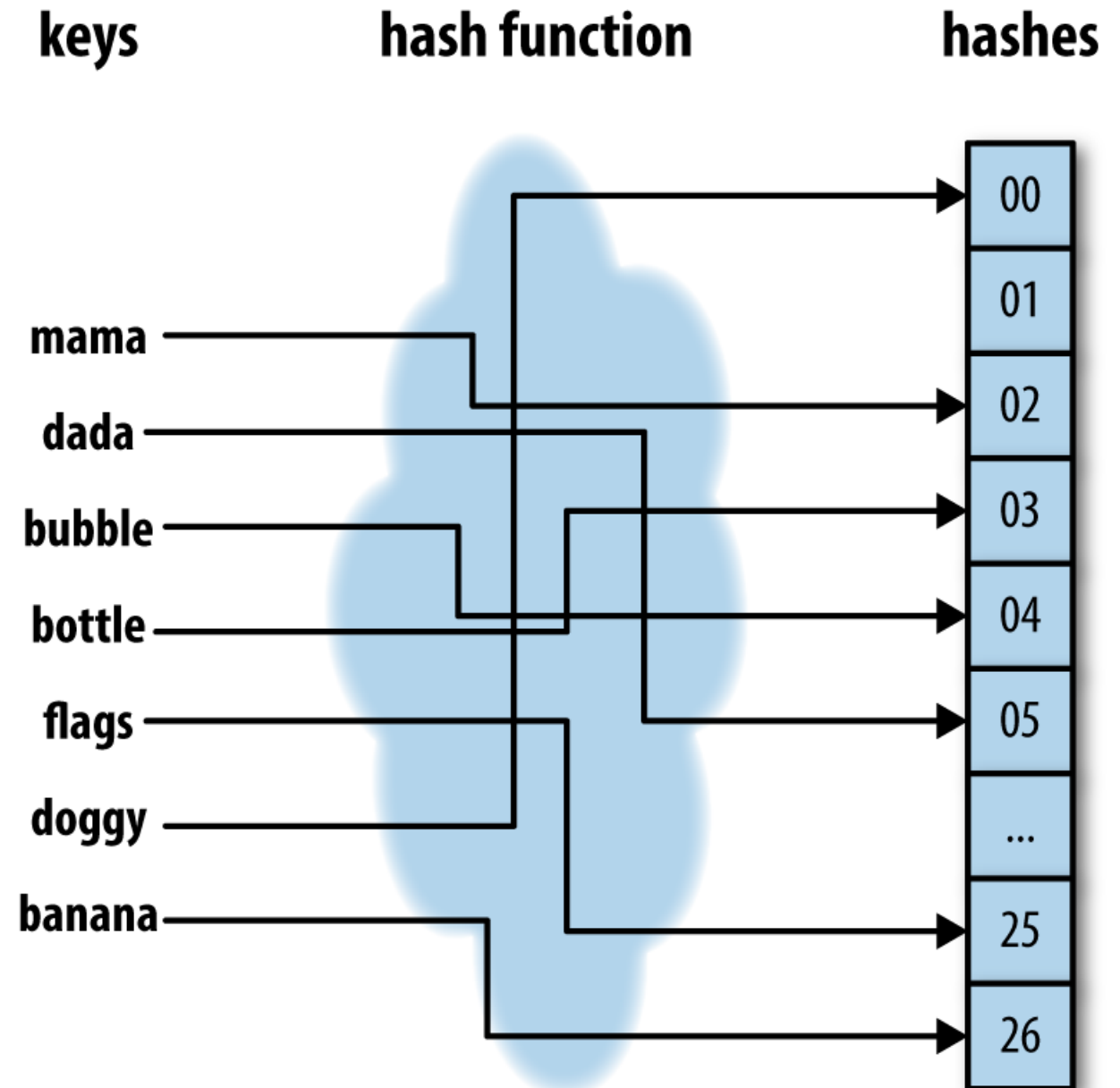
Encoding Categorical Features

Hashing Trick

Use a hash function to generate a hashed value of each category.

The hashed value will become the index of that category.

Can fix the number of encoded values for a feature in advance.



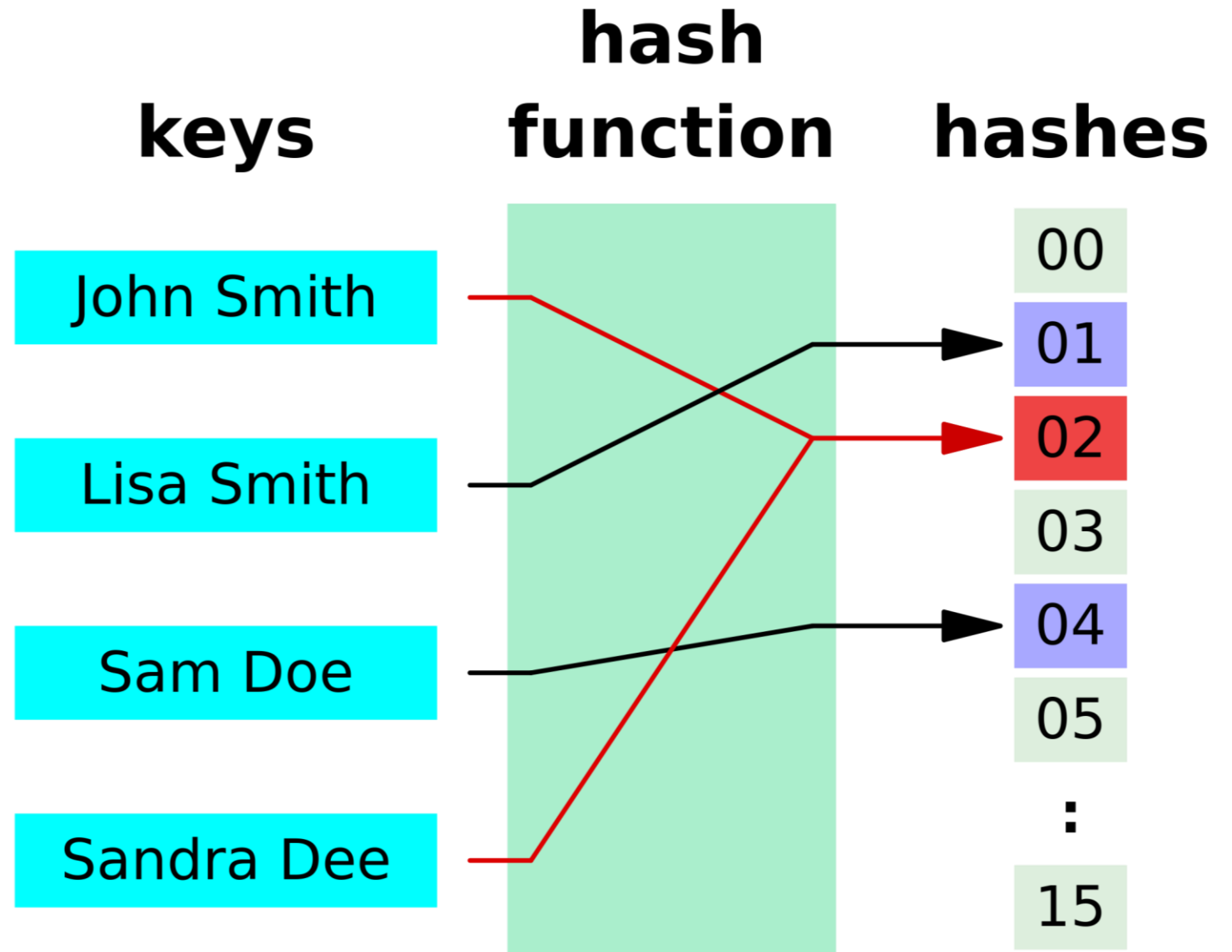
Hashing Trick

Collision

Two categories being assigned the same index, the collisions are random.

The impact of colliding hashed features is, fortunately, not that bad.

Hashing trick is often considered hacky by academics.



Feature Crossing

Technique to combine two or more features to generate new features. Useful for modeling the nonlinear relationships.

E.g. you suspect that there might be a nonlinear relationship between marital status and number of children.

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

Common Operations

There have been many techniques developed to streamline the process.

This list is nowhere near being comprehensive, but it does comprise some of the most common and useful operations.

- 1. Handling missing values**
- 2. Scaling**
- 3. Discretization**
- 4. Encoding categorical features**
- 5. Feature crossing**

ARTIFICIAL INTELLIGENCE

Hundreds of AI tools have been built to catch covid. None of them helped.

Some have been used in hospitals, despite not being properly tested. But the pandemic could help make medical AI better.

By Will Douglas Heaven

July 30, 2021



“Hundreds of AI Tools Have Been Built to Catch Covid. None of Them Helped.”

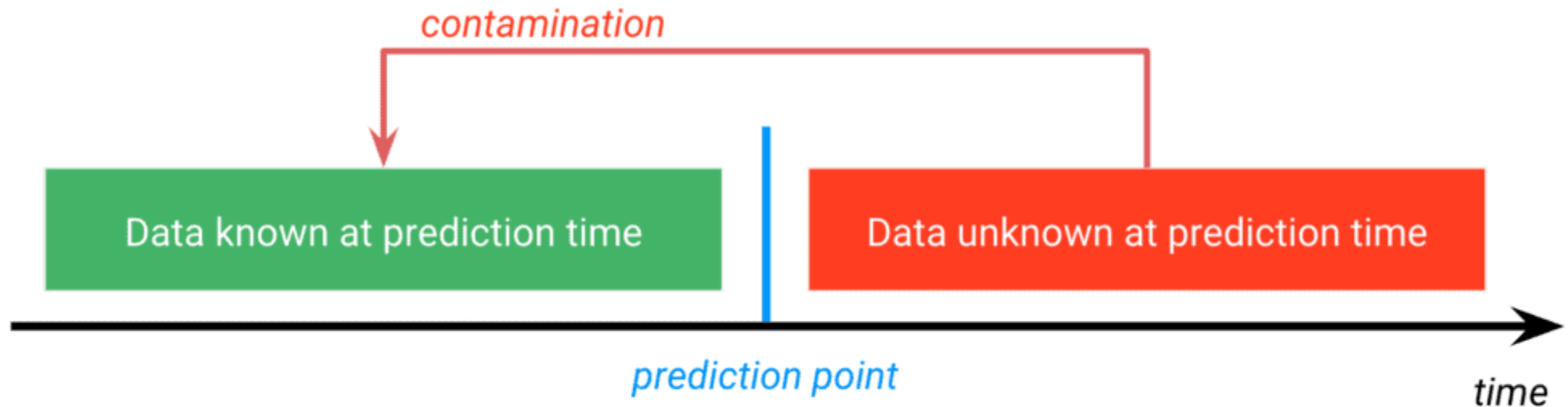
“Because patients scanned while lying down were more likely to be seriously ill, the model learned to predict serious covid risk from a person’s position.”

“Hundreds of AI Tools Have Been Built to Catch Covid. None of Them Helped.”

“... found to be picking up on the text font that certain hospitals used to label the scans. As a result, fonts from hospitals with more serious caseloads became predictors of covid risk.”

Data Leakage

The phenomenon when a form of the label “leaks” into the set of features used for making predictions, and this same information is not available during inference.



Data Leakage

Common Causes for Data Leakage

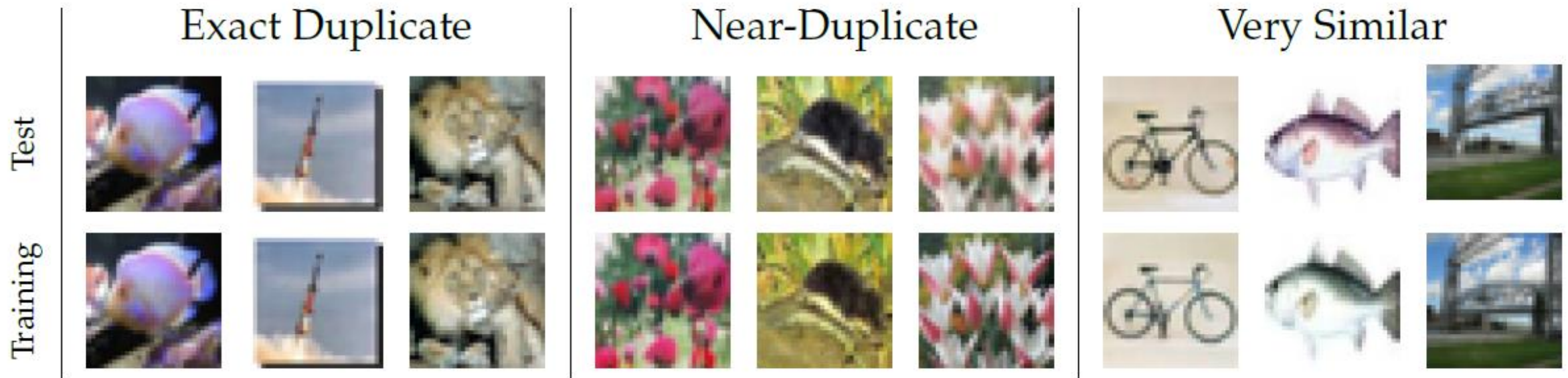
Splitting time-correlated data, scaling before splitting, statistics from test split, duplication before splitting, group leakage, data generation.

Train split					
Week 1	Week 2	Week 3	Week 4	Week 5	Valid split
X11	X21	X31	X41	X51	
X12	X22	X32	X42	X52	Test split
X13	X23	X33	X43	X53	
X14	X24	X34	X44	X54	
...	

Data Leakage

Common Causes for Data Leakage

Splitting time-correlated data, scaling before splitting, statistics from test split, duplication before splitting, group leakage, data generation.



Detecting Data Leakage

Data leakage can happen during many steps, from generating, collecting, sampling, splitting, and processing data to feature engineering.

- Measure the power of each feature
- Do removal studies of a feature
- Keep an eye on new features
- Be careful when looking at test data

Engineering Good Features

Generally, adding more features leads to better model performance.

However, more features doesn't always mean better model performance.

Might help models if the features that are not useful are removed.

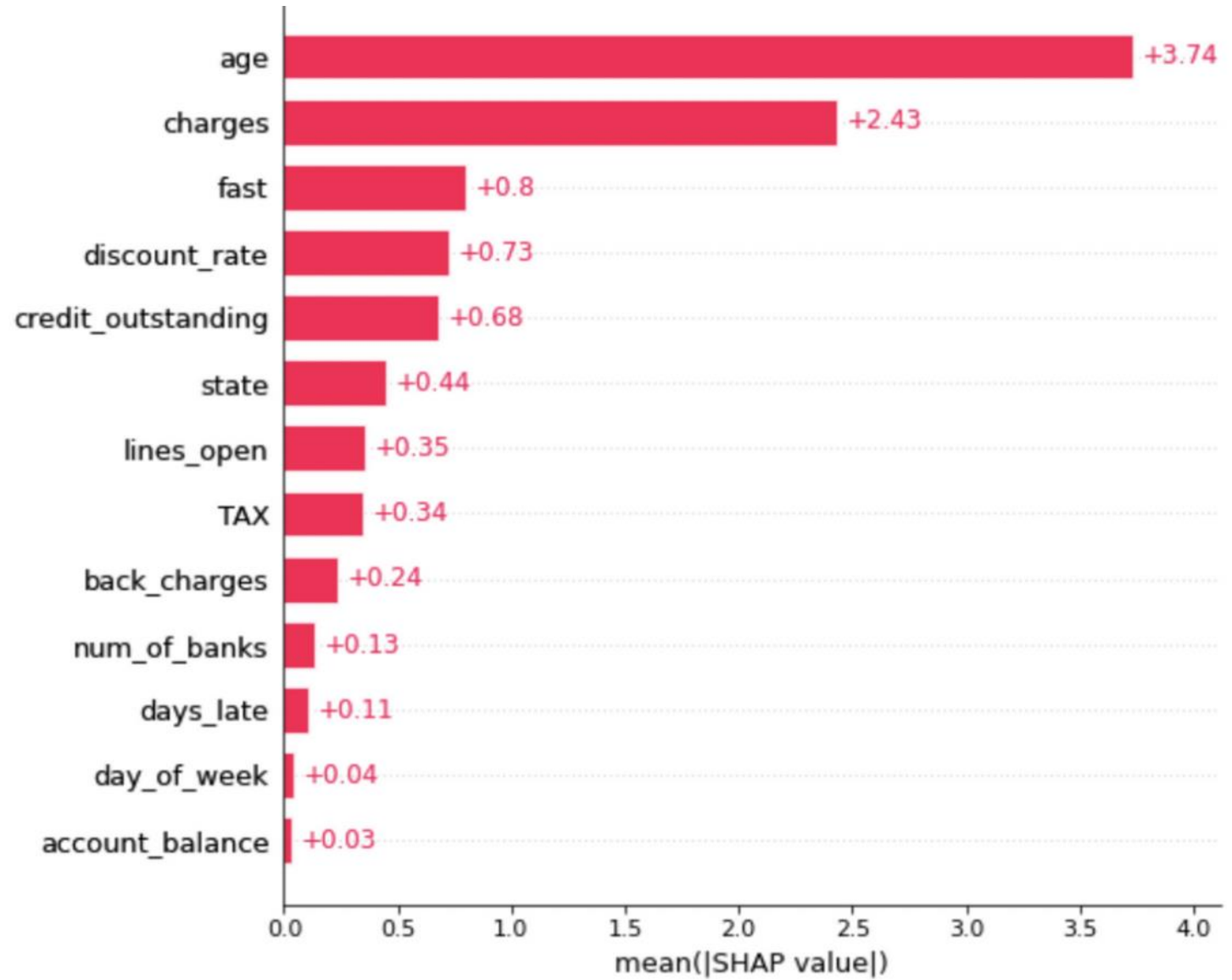
Having too many features can...

- Create opportunities for data leakage
- Cause overfitting
- Increase memory requirement
- Increase inference latency
- Become technical debts

Feature Importance

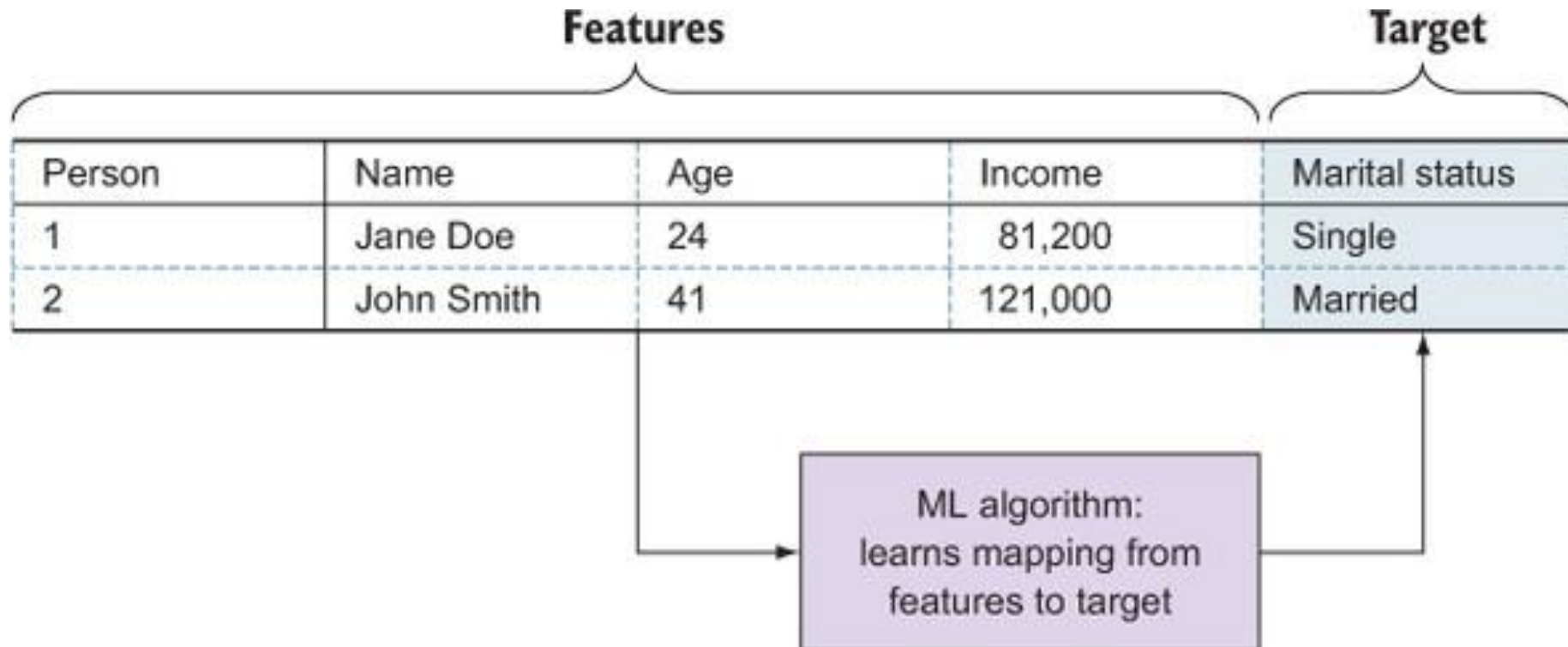
How much that model's performance deteriorates if that feature is removed from the model.

Often, a small number of features accounts for a large portion of your model's feature importance.



Feature Generalization

For the task of predicting whether a comment is spam, the identifier of each comment is not generalizable at all and shouldn't be used.



Best Practices for Feature Engineering

- Split data by time into train/valid/test splits.
- If you oversample your data, do it after splitting.
- Scale and normalize your data after splitting.
- Use statistics from only the train split.
- Understand your data.
- Keep track of your data.
- Understand feature importance to your model.
- Use features that generalize.
- Remove no longer useful features.

The End

Summary

- It's important to invest time and effort into feature engineering.
- Trying out different features and observing how they affect your models' performance.
- Feature engineering often involves subject matter expertise.
- We are never done with data and features.