Lecture 2
Data Preprocessing and Model Evaluation

CS 180 – Intelligent Systems

Dr. Victor Chen

Spring 2021

Data Preprocessing

What is Data?

Data include objects and their attributes

An object means an entity in the world

- Examples: person, transaction, image
- Object is also known as record, point, sample, or instance

An attribute is a property of an object

- Examples: name, date of birth, height, occupation are attributes of person.
- Attribute is also known as field, or feature

Attributes

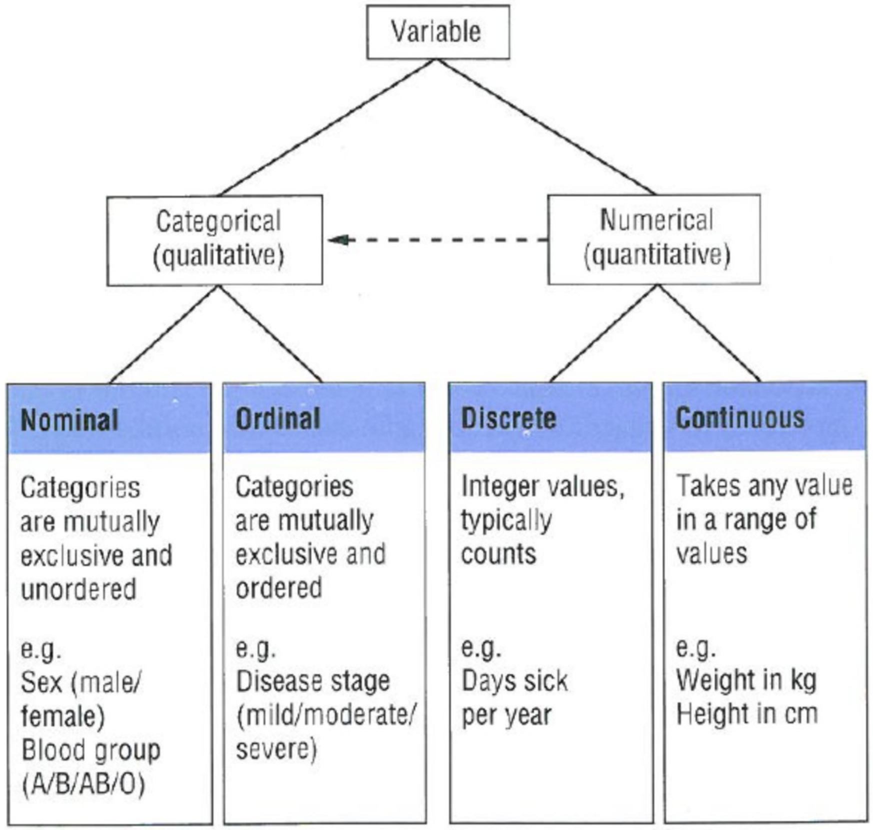
Objects

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Numeric and Categorical Data

Attributes can be:

- Numeric
  - Examples: dates, temperature, time, length
  - Discrete vs Continuous
- Categorical
  - Examples: eye color, rankings (e.g, good, fair, bad), height in {tall, medium, short}
  - Nominal (no order) vs Ordinal (ordered but not comparable)



If an attribute is categorical….

Data Encoding

If an attribute is categorical, we **must** convert it from categorical to numeric.

ID Number	Zip Code	Age	Marital Status	Income	Income Bracket	Refund
1129842	45221	55	Single	250000	High	No
2342345	45223	25	Married	30000	Low	Yes
1234542	45221	45	Divorced	200000	High	No
1243535	45224	43	Single	150000	Medium	No

Label Encoding

Label Encoding: Assign one unique number to each distinct value

original dataset

x_1	x_2	y
5	8	calabar
9	3	uyo
8	6	owerri
0	5	uyo
2	3	calabar
0	8	calabar
1	8	owerri

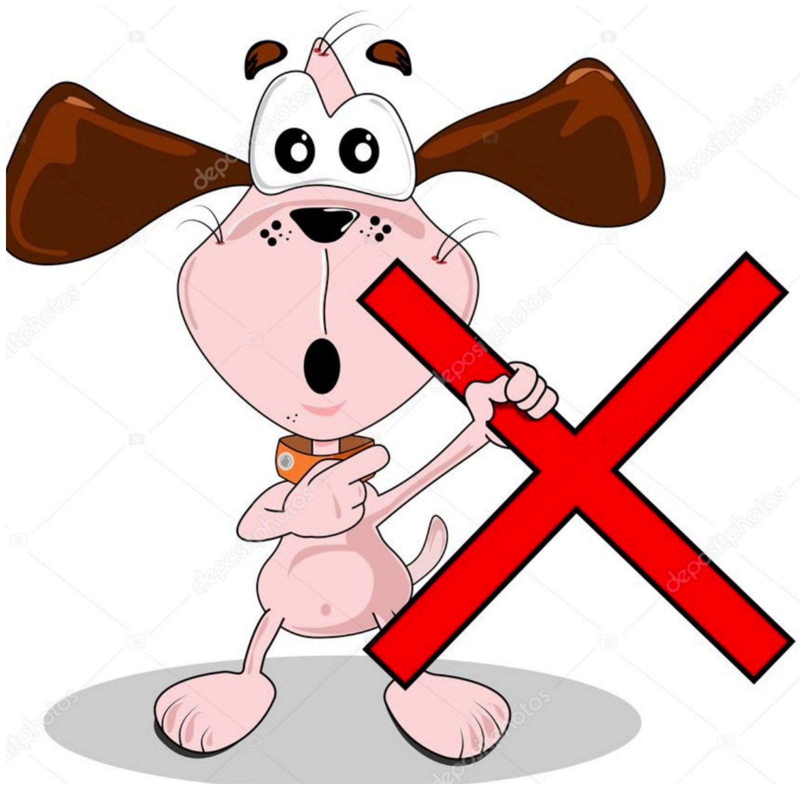
LabelEncoder

```
{
  "calabar" ---> 0
  "owerri"  ---> 1
  "uyo      ---> 2
}
```

dataset with encoded labels

x_1	x_2	y
5	8	0
9	3	2
8	6	1
0	5	2
2	3	0
0	8	0
1	8	1

Never use label encoding on input features



Data Encoding

How to convert the feature "Zip Code" from categorical to numeric?

ID Number	Zip Code	Age	Marital Status	Income	Income Bracket	Refund
1129842	45221	55	Single	250000	High	No
2342345	45223	25	Married	30000	Low	Yes
1234542	45221	45	Divorced	200000	High	No
1243535	45224	43	Single	150000	Medium	No

Data Encoding

One hot Encoding: For each distinct value, we create a new binary feature

ID	Zip 45221	Zip 45223	Zip 45224	Age	Single	Married	Divorced	Income	Refund
1129842	1	0	0	55	1	0	0	250000	0
2342345	0	1	0	25	0	1	0	30000	1
1234542	1	0	0	45	0	0	1	200000	0
1243535	0	0	1	43	1	0	0	150000	0

Thinking of records as vectors is very useful, which allows us to use linear algebra to process data.

Another example: One hot encoding

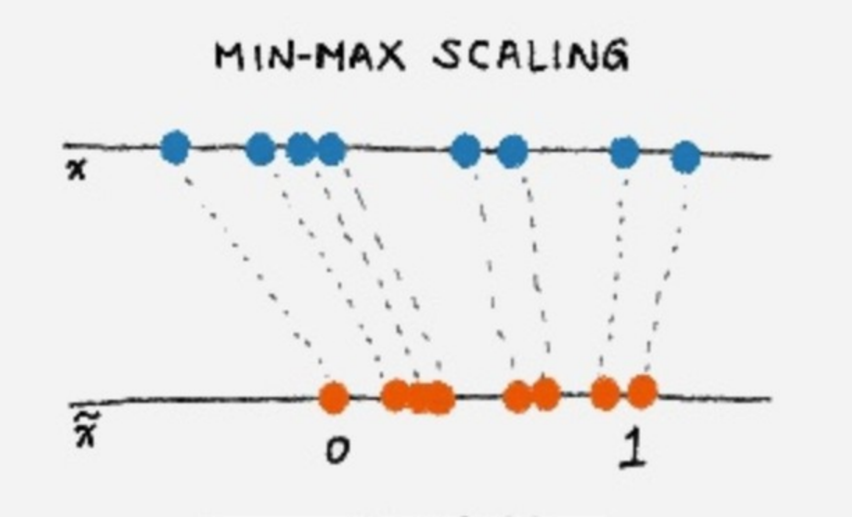
Sample	Category	Numerical
1	Human	1
2	Human	1
3	Penguin	2
4	Octopus	3
5	Alien	4
6	Octopus	3
7	Alien	4

Sample	Human	Penguin	Octopus	Alien
1	1	0	0	0
2	1	0	0	0
3	0	1	0	0
4	0	0	1	0
5	0	0	0	1
6	0	0	1	0
7	0	0	0	1

If an attribute is numeric….

Data normalization

If an attribute is numeric, we **must** normalize that attribute rather than using raw values



Normalization on numeric data

Different attributes take very different range of values.

We need to make them comparable

Temperature	Humidity	Pressure
30	0.8	90
32	0.5	80
24	0.3	95

Normalization – option 1

Divide raw values by the maximum value

Brings everything in the [0,1] range

Temperature	Humidity	Pressure
0.9375	1	0.9473
1	0.625	0.8421
0.75	0.375	1

new value = old value / max value in the column

Temperature	Humidity	Pressure
30	0.8	90
32	0.5	80
24	0.3	95

Normalization --- option 2

Subtract the minimum value and divide by the difference of maximum value and minimum value

Brings everything in the [0,1] range

Temperature	Humidity	Pressure
0.75	1	0.33
1	0.6	0
0	0	1

new value = (raw value – min column value) / (max col. value –min col. value)

Temperature	Humidity	Pressure
30	0.8	90
32	0.5	80
24	0.3	95

Normalization --- option 3

The standard score, *z*, of a raw value x is

$$z = \frac{x - \mu}{\sigma}$$

where:

μ is the mean.

σ is the standard deviation.

Standard scores are also called ***z*-scores**

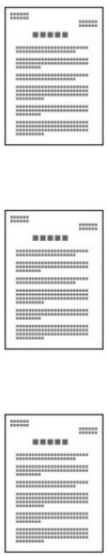
z score is negative when the raw value is below the mean, positive when above

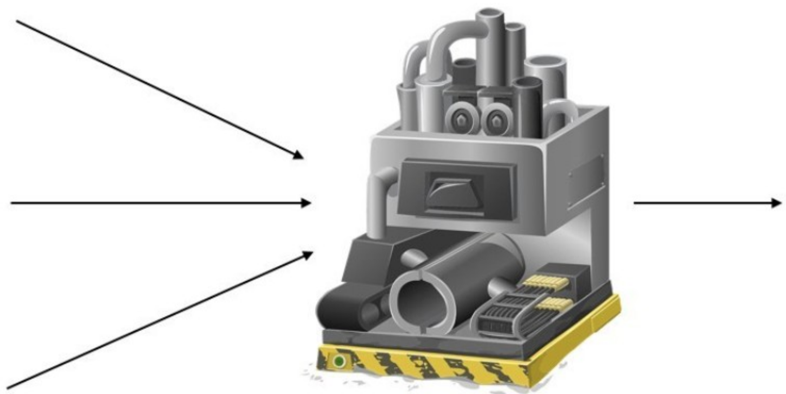
Natural Language Processing

Overview

- Bag-of-words (BoW) model
- TF-IDF model

BoW Model





	I	love	dogs	hate	and	knittin	is	my	hobby	passi
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

Document data

Doc Id	Words
1	the, dog, followed, the, cat
2	the, cat, chased, the, cat
3	the, man, walked, the, dog

Bag-of-words (BoW) model

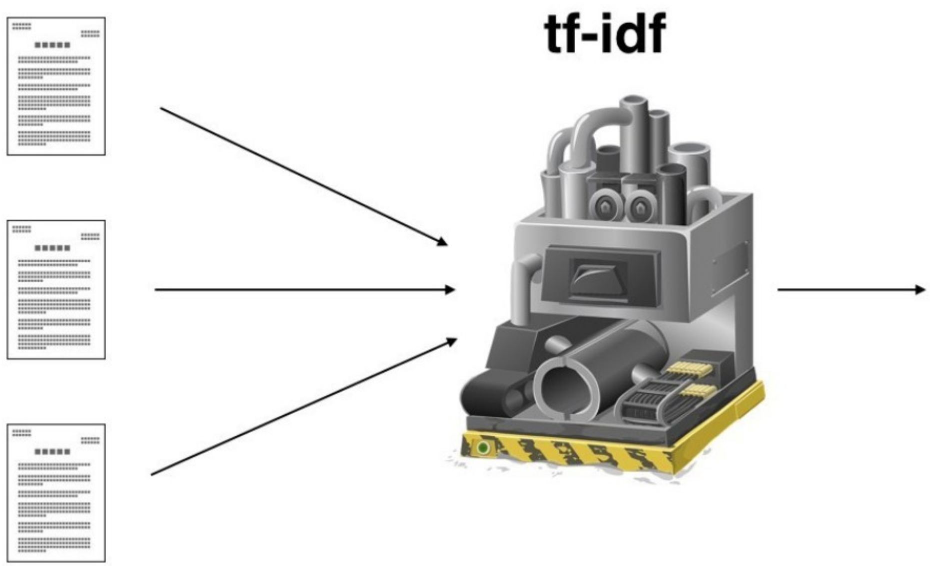
Create a feature for each unique word

- Each vector is defined over all possible words (vocabulary)
- The values are counts (number of times a word appears in the document, i.e., occurrence frequency)

Doc Id	Words
1	the, dog, follows, the, cat
2	the, cat, chases, the, cat
3	the, man, walks, the, dog

Doc Id	the	dog	follows	cat	chases	man	walks
1	2	1	1	1	0	0	0
2	2	0	0	2	1	0	0
3	1	1	0	0	0	1	1

Sparsity: Most entries are zero. Most documents contain few of the words



	I	love	dogs	hate	and	knittin	is	my	hobby	passi
Doc 1	0.18	0.48	0.18							
Doc 2	0.18		0.18	0.48	0.18	0.18				
Doc 3					0.18	0.18	0.48	0.95	0.48	0.48

TF-IDF model

Suppose we want to mine the business reviews of people on Yelp.

Yelp screenshots

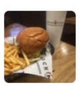
Rock-N-Fire Claimed

202 reviews

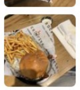
$$ · Pizza, American (Traditional), Tapas/Small Plates Edit

Write a Review | Add Photo | Share | Save

Review Highlights



"I ordered the **Ringer burger** with sweet potato fries and my sister ordered the Kickin' Chicken pizza." in 6 reviews



"First time here, got a burger with **garlic fries** and the wife got a black bean burger." in 13 reviews

"Friendly service and well trained staff, fresh ingredients and a **clean environment**. Really tasty beers on draft as well." in 5 reviews

Glenda C.
Newcastle, CA
223 friends
36 reviews
40 photos

7/15/2019
7 photos

Scott took me on a date to Rock-N-Fire.
The pizzas were the... BOMB!!!
Trey & Mike made us our very own custom pizzas.
The salad & pizza were beautifully created
The owner, Mike was delightful & the atmosphere is very inviting & fun!
Be sure to visit & have a delightful experience.
We'll definitely be back!!!

Useful | Funny | Cool 1

Matt I.
Granite Bay, CA
0 friends
11 reviews
10 photos

5/23/2019
1 photo

Customer Service: 3.5/5
Food: 2.5/5
Atmosphere: 3/5
Cleanliness: 3.5/5

Why I gave the food a 2.5/5? We ordered burgers that sounded better than they tasted, and additionally had to purchase fries to an $8 burger. Not that there's a problem with this concept, but a single fast-food burger and fries came out to $12, two for $25. The sweet potato fries we received were also burnt. In my book a 2.5 is still average.

Data collection

Collect all reviews for restaurants in NY in Yelp

- Yelp API gives you each review in JSON format
- https://www.yelp.com/developers


```
{"votes": {"funny": 0, "useful": 2, "cool": 1},
 "user_id": "Xqd0DzHaiyRqVH3WRG7hzg",
 "review_id": "15SdjuK7DmYqUAj6rjGowg",
 "stars": 5, "date": "2007-05-17",
 "text": "I heard so many good things about this place so I was pretty juiced to
try it.  I'm from Cali and I heard Shake Shack is comparable to IN-N-OUT and I
gotta say,  Shake Shake wins hands down.    Surprisingly, the line was short and
we waited about 10 MIN. to order.  I ordered a regular cheeseburger, fries and a
black/white shake.  So yummerz.   I love the location too!  It's in the middle of
the city and the view is breathtaking.   Definitely one of my favorite places to
eat in NYC.",
 "type": "review",
 "business_id": "vcNAWiLM4dR7D2nwwJ7nCA"}
```

Flat text data (one file per business)

I heard so many good things about this place so I was pretty juiced to try it. I'm from Cali and I heard Shake Shack is comparable to IN-N-OUT and I gotta say, Shake Shake wins hands down. Surprisingly, the line was short and we waited about 10 MIN. to order. I ordered a regular cheeseburger, fries and a black/white shake. So yummerz. I love the location too! It's in the middle of the city and the view is breathtaking. Definitely one of my favorite places to eat in NYC.

I'm from California and I must say, Shake Shack is better than IN-N-OUT, all day, err'day.

Would I pay $15+ for a burger here? No. But for the price point they are asking for, this is a definite bang for your buck (though for some, the opportunity cost of waiting in line might outweigh the cost savings) Thankfully, I came in before the lunch swarm descended and I ordered a shake shack (the special burger with the patty + fried cheese & portabella topping) and a coffee milk shake. The beef patty was very juicy and snugly packed within a soft potato roll. On the downside, I could do without the fried portabella-thingy, as the crispy taste conflicted with the juicy, tender burger. How does shake shack compare with in-and-out or 5-guys? I say a very close tie, and I think it comes down to personal affliations. On the shake side, true to its name, the shake was well churned and very thick and luscious. The coffee flavor added a tangy taste and complemented the vanilla shake well. Situated in an open space in NYC, the open air sitting allows you to munch on your burger while watching people zoom by around the city. It's an oddly calming experience, or perhaps it was the food coma I was slowly falling into. Great place with food at a great price.

First cut

Remove punctuation, make into lower case, clear white spaces,

For each business, break into words, keep the most popular words

the 27514	the 16710	the 16010	the 14241
and 14508	and 9139	and 9504	and 8237
i 13088	a 8583	i 7966	a 8182
a 12152	i 8415	to 6524	i 7001
to 10672	to 7003	a 6370	to 6727
of 8702	in 5363	it 5169	of 4874
ramen 8518	it 4606	of 5159	you 4515
was 8274	of 4365	is 4519	it 4308
is 6835	is 4340	sauce 4020	is 4016
it 6802	burger 432	in 3951	was 3791
in 6402	was 4070	this 3519	pastrami 3748
for 6145	for 3441	was 3453	in 3508
but 5254	but 3284	for 3327	for 3424
that 4540	shack 3278	you 3220	sandwich 2928
you 4366	shake 3172	that 2769	that 2728
with 4181	that 3005	but 2590	but 2715
pork 4115	you 2985	food 2497	on 2247
my 3841	my 2514	on 2350	this 2099
this 3487	line 2389		
wait 3184	this 2242		
not 3016	fries 2240	chicken 2220	not 1655
we 2984	on 2204	with 2195	your 1622
at 2980	are 2142	rice 2049	so 1610
on 2922	with 2095	so 1825	have 1585

What can you observe?

First cut

- Remove punctuation, make into lower case, clear white spaces,
- For each business, break into words, keep the most popular words

```
the 27514
and 14508
i 13088
a 12152
to 10672
of 8702
ramen 8518
was 8274
is 6835
it 6802
in 6402
for 6145
but 5254
that 4540
you 4366
with 4181
pork 4115
my 3841
this 3487
wait 3184
not 3016
we 2984
at 2980
on 2922
```

```
the 16710
and 9139
a 8583
i 8415
to 7003
in 5363
it 4606
of 4365
is 4340
burger 432
was 4070
for 3441
but 3284
shack 3278
shake 3172
that 3005
you 2985
my 2514
line 2389
this 2242
fries 224
on 2204
are 2142
with 2095
```

```
the 16010
and 9504
i 7966
to 6524
a 6370
it 5169
of 5159
is 4519
sauce 4020
in 3951
this 3519
was 3453
for 3327
you 3220
that 2769
but 2590
food 2497
on 2350
chicken 2220
with 2195
rice 2049
so 1825
```

```
the 14241
and 8237
a 8182
i 7001
to 6727
of 4874
you 4515
it 4308
is 4016
was 3791
pastrami 3748
in 3508
for 3424
sandwich 2928
that 2728
but 2715
on 2247
this 2099
your 1622
so 1610
have 1585
```

Most frequent words are stop words

Second cut

After removing stop words…

ramen 8572
pork 4152
wait 3195
good 2867
place 2361
noodles 2279
ippudo 2261
buns 2251
broth 2041
like 1902
just 1896
get 1641
time 1613
one 1460
really 1437
go 1366
food 1296
bowl 1272
can 1256
great 1172
best 1167

burger 4340
shack 3291
shake 3221
line 2397
fries 2260
good 1920
burgers 1643
wait 1508
just 1412
cheese 1307
like 1204
food 1175
get 1162
place 1159
one 1118
long 1013
go 995
time 951
park 887
can 860
best 849

sauce 4023
food 2507
cart 2239
chicken 2238
rice 2052
hot 1835
white 1782
line 1755
good 1629
lamb 1422
halal 1343
just 1338
get 1332
one 1222
like 1096
place 1052
go 965
long 792
people 790

pastrami 3782
sandwich 2934
place 1480
good 1341
get 1251
katz's 1223
just 1214
like 1207
meat 1168
one 1071
deli 984
best 965
go 961
ticket 955
food 896
sandwiches 813
can 812
time 662

What can you observe?

Second cut

After removing stop words…

ramen 8572
pork 4152
wait 3195
good 2867
place 2361
noodles 2279
ippudo 2261
buns 2251
broth 2041
like 1902
just 1896
get 1641
time 1613
one 1460
really 1437
go 1366
food 1296
bowl 1272
can 1256
great 1172
best 1167

burger 4340
shack 3291
shake 3221
line 2397
fries 2260
good 1920
burgers 1643
wait 1508
just 1412
cheese 1307
like 1204
food 1175
get 1162
place 1159
one 1118
long 1013
go 995
time 951
best 849

sauce 4023
food 2507
cart 2239
chicken 2238
rice 2052
hot 1835
white 1782
line 1755
good 1629
lamb 1422
halal 1343
just 1338
get 1332
one 1222
like 1096
place 1052
go 965
can 878
long 792
people 790

pastrami 3782
sandwich 2934
place 1480
good 1341
get 1251
katz's 1223
just 1214
like 1207
meat 1168
one 1071
deli 984
best 965
go 961
ticket 955
food 896
sandwiches 813
can 812
beef 768
time 662

Commonly used words in reviews, not so interesting

Importance measure: Term Frequency (TF)

TF(w,d): term frequency of word w in document d

- A measure of importance of a word w for a document d
- TF(w,d): = (Number of times w appears in d) / (Total number of terms in d).

Uniqueness measure: Inverse Document Frequency (IDF)

Document Frequency of a word , : fraction of documents that contain word .

: num of docs that contain word
: total number of documents in dataset

Inverse Document Frequency of a word :

TF-IDF

For document analysis, we are more interested in the words that are not only important for the document, but also unique to the document.

TF(w,d): term frequency of word w in document d

- A measure of importance of the word w for the document d

IDF(w): inverse document frequency

- A measure of the uniqueness of the word w

$$\text{TF-IDF}(w,d) = \text{TF}(w,d) \times \text{IDF}(w)$$

TF-IDF: put everything together

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

TF-IDF: An Example

Document contains 100 words

- Word "apple" appears 10 times in
- Word "orange" appears 20 times in

We have documents in total

- Word "apple" only appears in document
- Word "orange" appears in all 20 documents

$$tf-idf(\text{“}apple\text{”}, d_1) = \frac{10}{100} \times \log_2 \frac{20}{1} =$$

0.432

$$tf-idf(\text{“}orange\text{”}, d_1) = \frac{20}{100} \times \log_2 \frac{20}{20} =$$

0

Third cut

Order all the words by TF-IDF per document

ramen 3057.4176194

brulee 201.1797390

fries 806.085373301

lamps 99.011158998

lamb 985.655290756

moma 48.1589121730

pastrami 1931.94250908298 6

508209 4
888 2
700399 2
1580035 4
830666 1
89887 1
07471 2
72916 4
8853014 6
462458 1
12713779 3
6609 7
39524 1
5707218 8
5389878 4
13392 7
258549 3
5021678 1
488967607 2
742696 1
649702 1
8045519 2

matzoh 108.22149937072 1

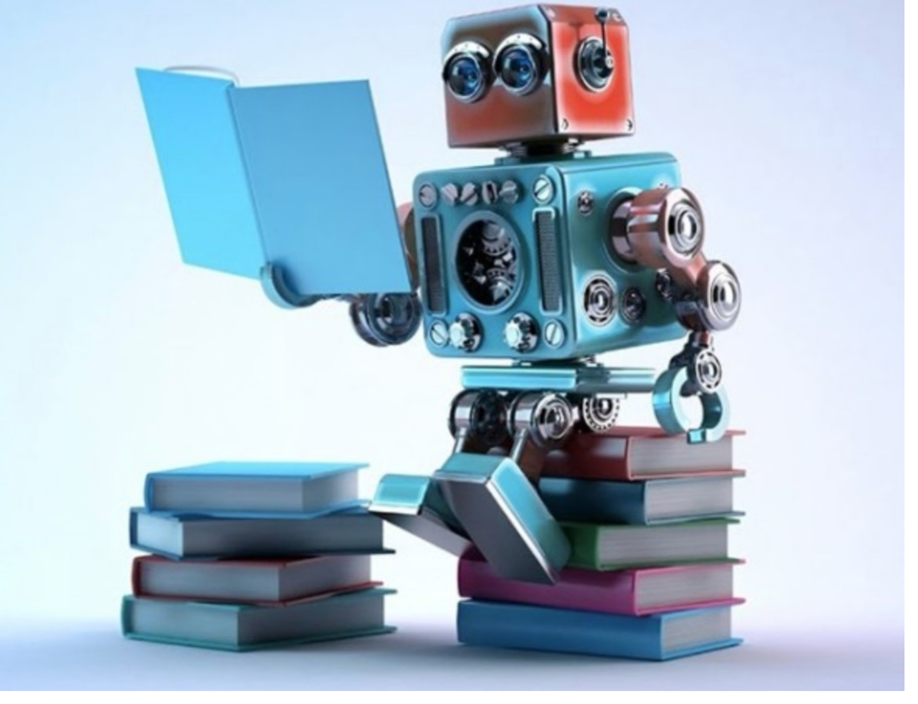
Done!

Now, we get the TF-IDF vector for each document (restaurant), which are ready to be sent to models as input

Modeling and evaluation

Machine learning

- A study on getting a computer to finish a task without explicitly programming it



Learning types

Unsupervised
(data has no
labels)

Supervised
(all data are
labeled)

Machine Learning in a nutshell

	Supervised Learning	Unsupervised Learning
Continuous Output	Regression (linear regression, neural networks,…)	Dimensionality reduction
Discrete Output	Classification (KNN, SVM, decision tree, Bayes classifier, logistic regression, neural networks…)	Clustering (k-means, hierarchical clustering, DBSCAN, GMM…)

Classification

Learn a model from labeled data and make prediction on labels.

– Predict the species of Iris flower (**setosa**, **virginica** and **versicolor**)

Classification

Attributes

sepal_length	sepal_width	petal_length	petal_width	Iris_class
5	2	3.5	1	versicolor
6	2.2	4	1	versicolor
6.2	2.2	4.5	1.5	versicolor
6	2.2	5	1.5	virginica
4.5	2.3	1.3	0.3	setosa
5.5	2.3	4	1.3	versicolor
6.3	2.3	4.4	1.3	versicolor
5	2.3	3.3	1	versicolor
4.9	2.4	3.3	1	versicolor
5.5	2.4	3.8	1.1	versicolor
5.5	2.4	3.7	1	versicolor
5.6	2.5	3.9	1.1	versicolor
6.3	2.5	4.9	1.5	versicolor
5.5	2.5	4	1.3	versicolor
5.1	2.5	3	1.1	versicolor
4.9	2.5	4.5	1.7	virginica
6.7	2.5	5.8	1.8	virginica
5.7	2.5	5	2	virginica
6.3	2.5	5	1.9	virginica
5.7	2.6	3.5	1	versicolor
5.5	2.6	4.4	1.2	versicolor
5.8	2.6	4	1.2	versicolor

Data point /example

Numerical value

Categorical value

Steps in classification

Define the problem

- What you are trying to predict?

Identify **input features**

- Find the features that help to discriminate between the classes

Identify **output labels**

Decide on which model to train

- What is the right model for your problem?

Train the model on training data

Test the model on test data

Any model

$$y = f_{\Theta}(\mathbf{x})$$

Output(label) function Input (features)

Collecting data for computer vision

apple

pear

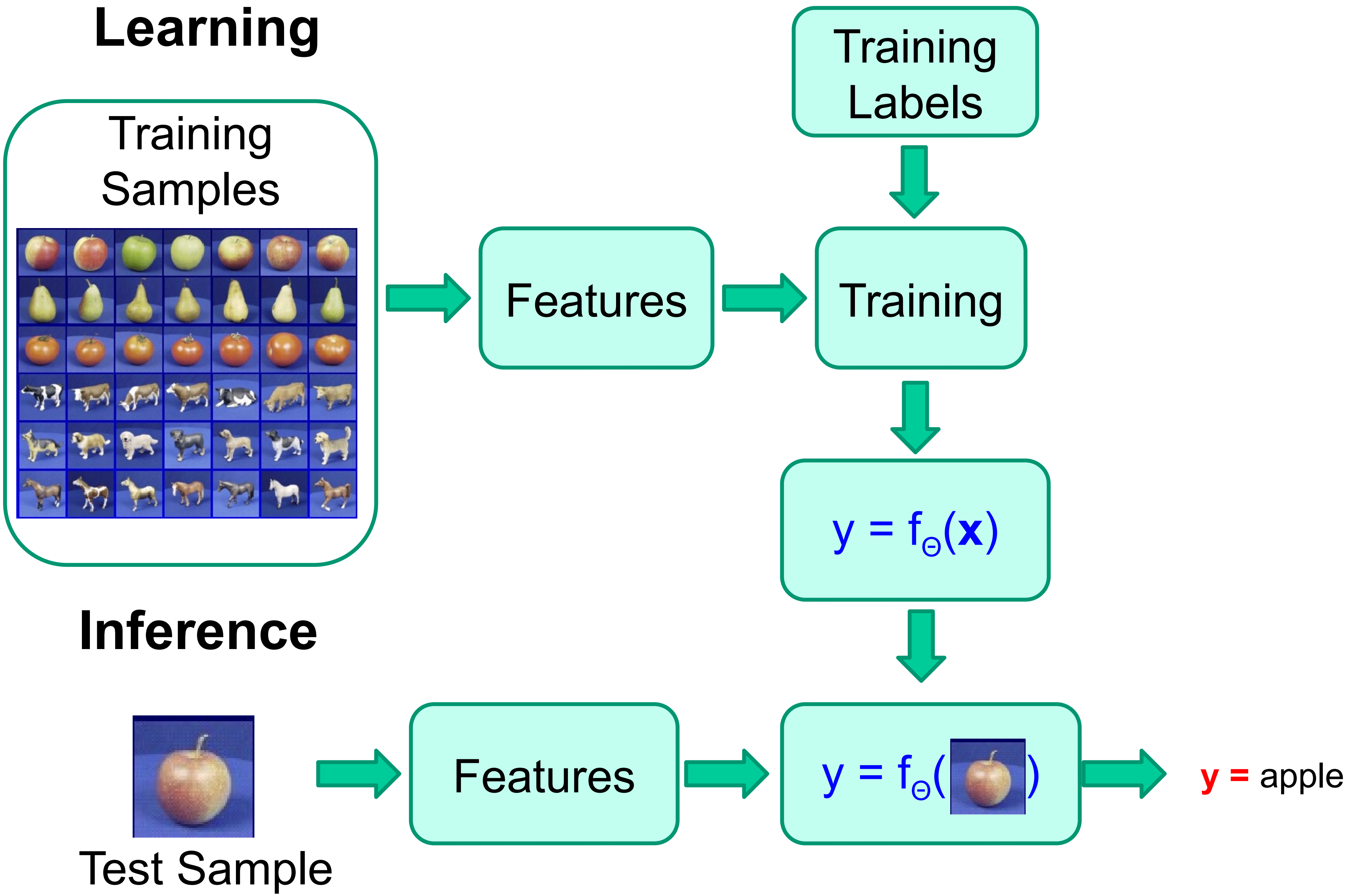
tomato

cow

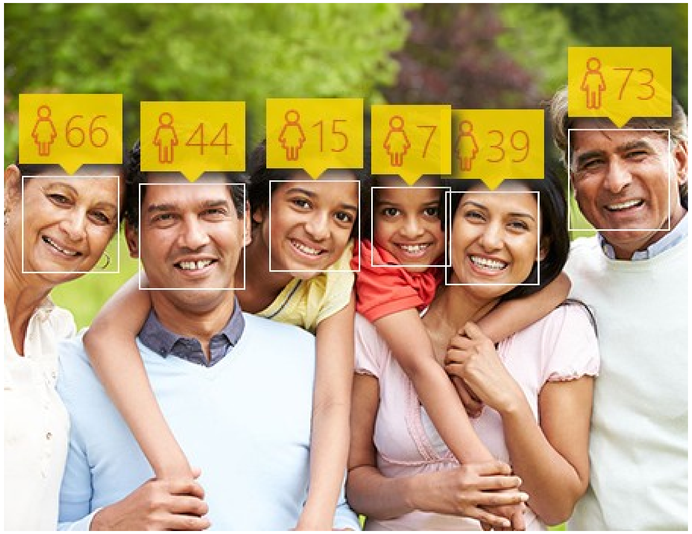
dog

horse

Machine learning pipeline



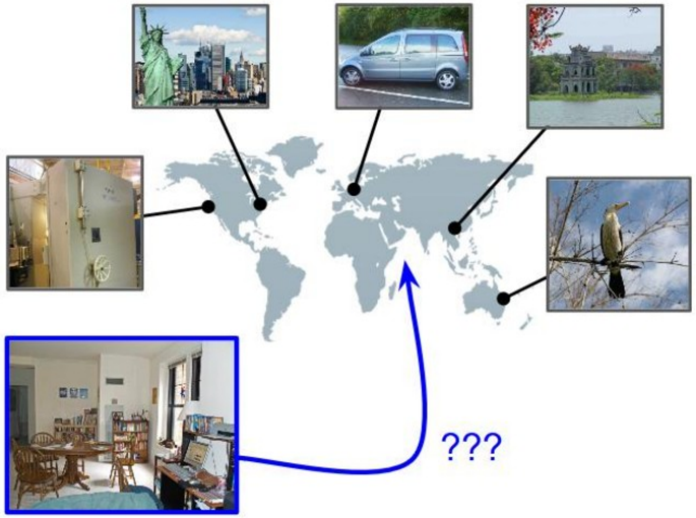
Regression



Age estimation

When was that made?

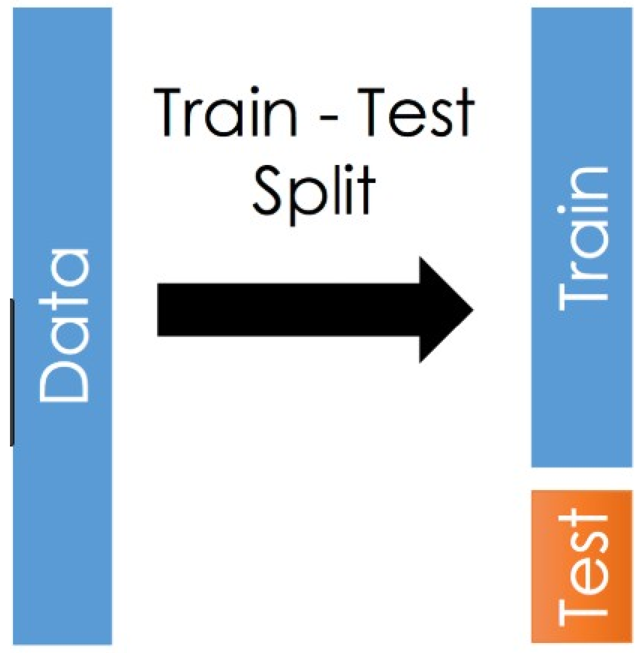
IM2GPS



Model evaluation

Split data into two separate parts

- Learn *parameters* on the *training set*
- Evaluate performance on the *test set*
- ~70% of records used for training, ~30% for testing

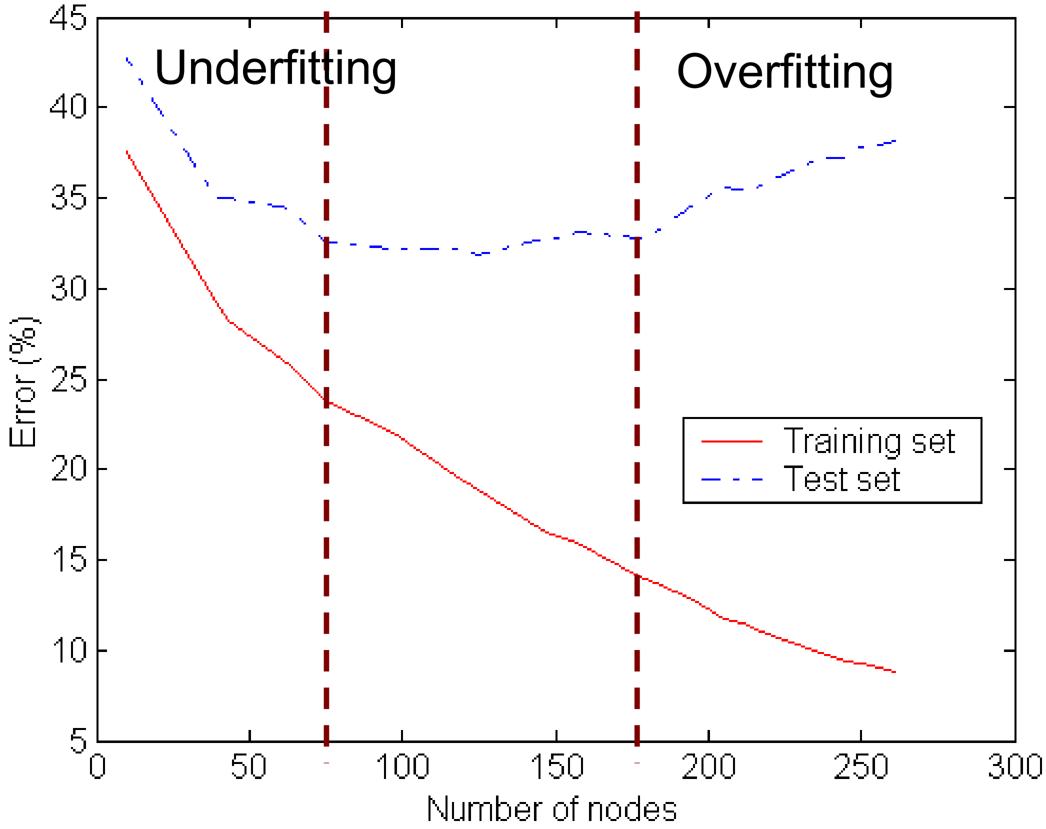


ML model evaluation

The ultimate goal of evaluation should be:

MAXIMIZE the ability of the model to predict **new (out-of-sample) data.**

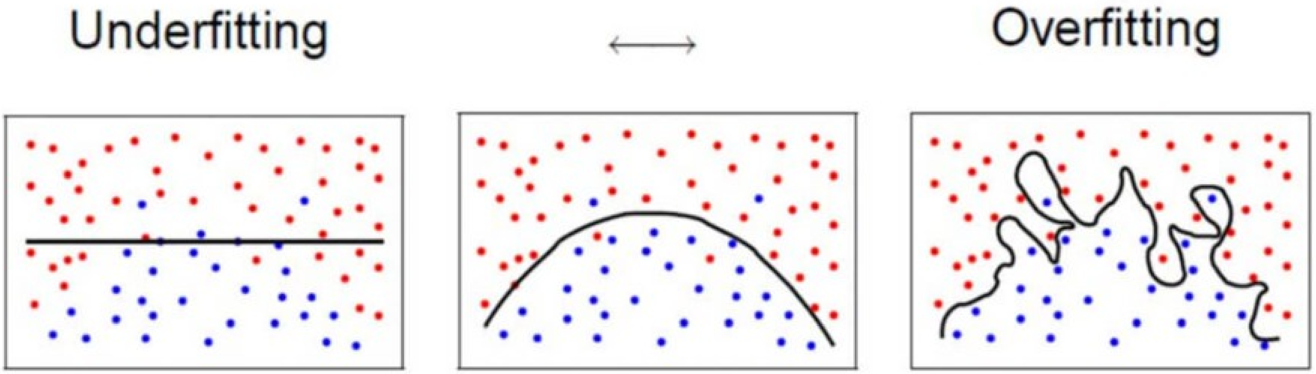
Underfitting and Overfitting



Underfitting: when model is too simple, both training and test errors are large

Overfitting: when model is too complex it models the details of the training set and fails to generalize well on the test set

Underfitting and Overfitting



How to avoid underfitting and overfitting?

Train model *parameters* on *training set*
Test model *performance* on *test set*

Stop training once the test error goes up!

What metrics should be used to measure test error?

Metric used to evaluate model on test data

- Classification
  - Precision
  - Recall
  - F1-score
  - ROC curve

- Regression
  - RMSE (Root Mean Squared Error)
  - R2 score

Precision, Recall, F1-score

Precision: Out of records labeled as positive, how many are actually positive?

Recall: How many positive records are labeled correctly?

F-measure (F1-score): It combines precision and recall and can be used as a single summary number for model quality

$$\text{F-measure (F)} = \frac{2rp}{r + p}$$

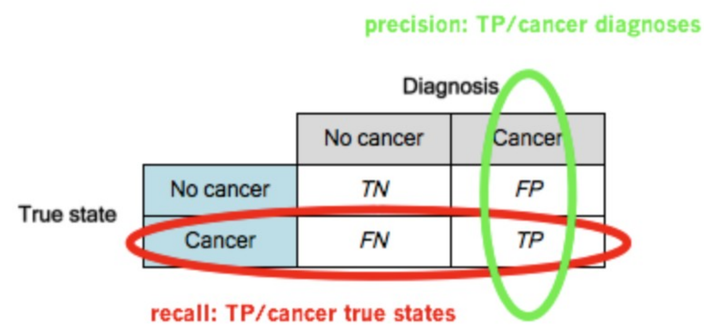
Confusion matrix (binary problem)

		PREDICTED CLASS	
		Class=No	Class=Yes
ACTUAL CLASS	Class=No	a (TN)	b (FP)
	Class=Yes	c (FN)	d (TP)

$$\text{Precision (p)} = \frac{a}{a+c} = \frac{TP}{TP+FP}$$

$$\text{Recall (r)} = \frac{a}{a+b} = \frac{TP}{TP+FN}$$

$$\text{F-measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c} = \frac{2TP}{2TP+FP+FN}$$



Confusion matrix

Confusion matrix about a model used to predict whether a tumor is malignant (NO) or benign (YES)

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

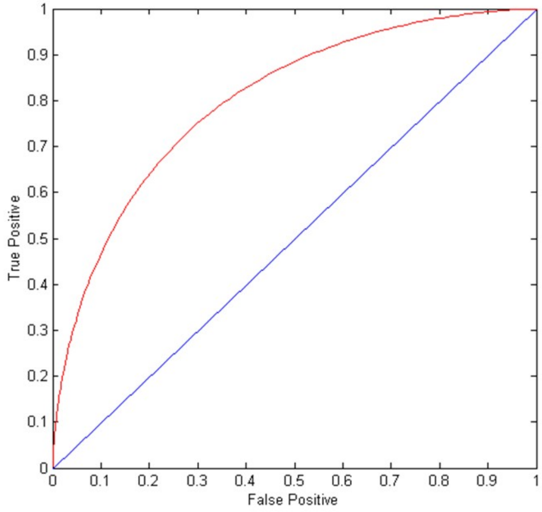
ROC Curve

ROC curve plots TPR (true positive rate) (on the y-axis) against FPR (false positive rate) (on the x-axis)

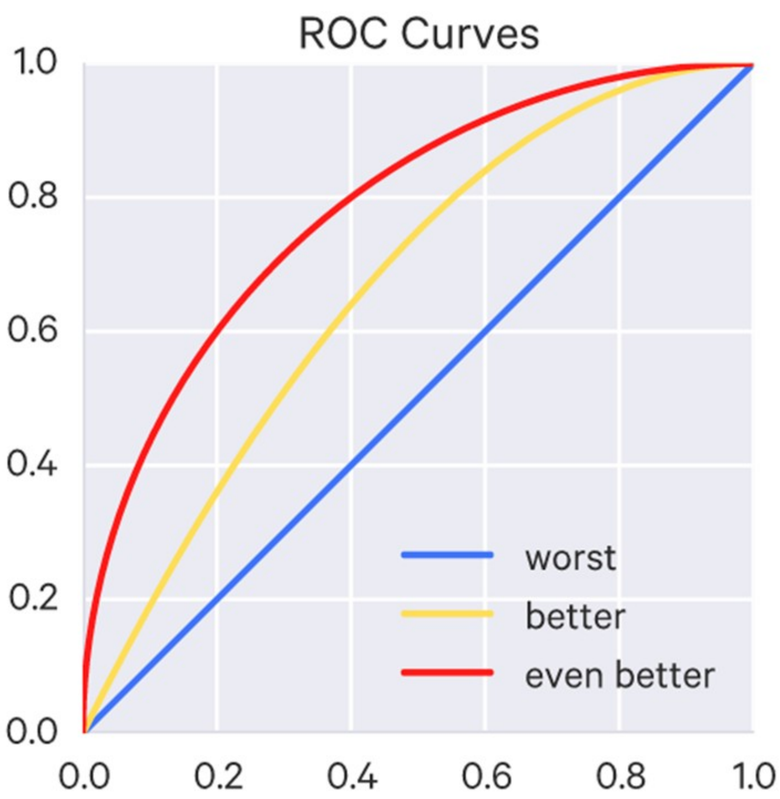
$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

		PREDICTED CLASS	
		Class=No	Class=Yes
ACTUAL CLASS	Class=No	a (TN)	b (FP)
	Class=Yes	c (FN)	d (TP)



Using ROC for Model Comparison



- Area Under Curve (AUC) determines which models preforms best
  - Ideal: Area = 1
  - Random guess (= Diagonal line) :
    - Area = 0.5