# Notes: Feature Engineering

General references:

1. https://www.kaggle.com/notebooks?sortBy=voteCount&group=everyone&pageSize=20&datasourceType=competitions
2. https://www.kaggle.com/shivamb/extensive-text-data-feature-engineering
3. https://www.kaggle.com/sudalairajkumar/getting-started-with-text-preprocessing
4. https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-te
5. Textbook, Section 25.2

**Important:** Keep in mind the differences between bayes error, approximation error, and estimation error throughout this discussion.

## 1 Text

**Problem 1.** Why is text data difficult?

almost always has high bayes err.

I never said she stole my money

if we had audio,

bayes error ↓ because we could detect the emphasis

**Problem 2.** What is the difference between a *distributed* encoding and a *1-hot* encoding?

encoding a function

$f : \text{string} \to \mathbb{R}^d$

transformers generates dist. encode.

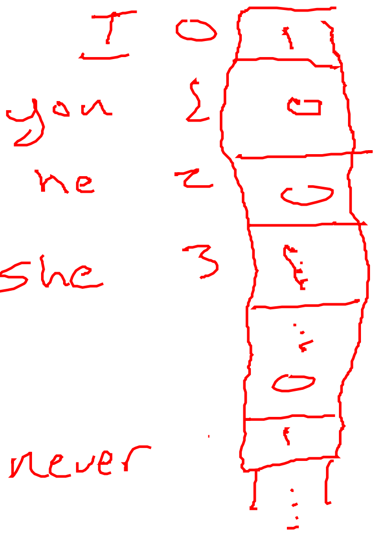
- dense vectors
- d is small 512 - 1024
- hard to interpret

1-hot encoding

- sparse vector
- d is large $10^6 - 10^9 - 10^{12}$
- easy to interpret

vocab dictionary

| | |
|---|---|
| I | 0 |
| never | 57 |
| said | 201 |
| she | 3 |
| stole | 88 |
| ⋮ | |



**Problem 3.** A 1-hot encoding of words is sometimes called a *bag of words*. What are it's limitations?

1. Large dimensionality

computational - important to keep sparse; sometimes, have to rep. as dense

statistical -

large estimation error

need to use models with low VCdim

using the 1-hot encoding will ↑ your bayes error

2. The context problem (i.e., not bijective)

I hate cats and love dogs

I love cats and hate dogs

both sentences have same 1-hot encoding

3. The phrase problem

I love New York

I love my new york terrier

4. The synonym problem

can't

cannot

can_not

New York

New York City

Big Apple

home

homonym prob

can - noun

can - verb

5. The punctuation problem

   See: `https://digitalsynopsis.com/tools/punctuation-marks-importance-rules-usage/`

6. The tokenization problem

   token = word

   ☺

   url

   spacy

| | |
|---|---|
| English | I love data mining |
| Spanish | Me encanta la minería de datos |
| Chinese | 我喜歡數據挖掘 |
| Vietnamese | Tôi thích khai thác dữ liệu |

7. The compound word problem

| | |
|---|---|
| English | Danube steam ship company captain |
| German | Donaudampfschiffahrtsgesellschaftskapit |

should we break up compound words?

8. The conjugation problem

encantar - Spanish

> 200 different forms

love, loved, loves, loving

9. The Unicode problemS

   See the video "Unicode and Python: the absolute minimum you need to know": `https://www.youtube.com/watch?v=oXVmZGN6plY`

encantar

encanta

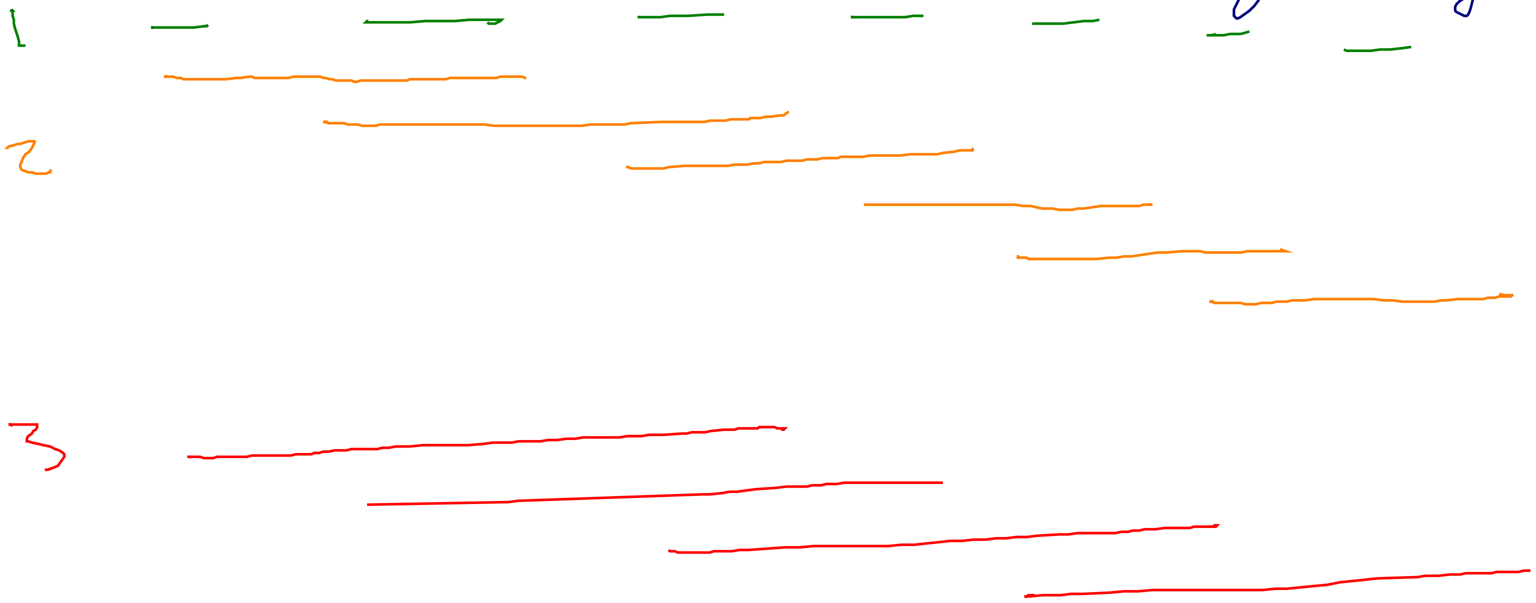
encantaste

- char -level n-gram
- syl. -level n-gram

**Problem 4.** What are $n$-grams? What are the tradeoffs of using $n$-grams?

standard 1-hot encoding uses 1-grams

I never said she stole my money



when using n-grams, m-grams

$1 \leq m \leq n$

vocab size, d words

1 grams: $d$

2 grams: $d^2$

3 grams: $d^3$

memory reqts

vectors get larger

$\Rightarrow$ ↑ VC dim

↑ estimation err.

Bayes error ↓

**Problem 5.** What is lemmatization? What are the tradeoffs of using lemmatization?

great for foreign langs.

converts words into standard form

love, loves, loved, loving

⇓

love

encantar, encanta, encantaste

⇓

encantar

reduces size of vocabulary

est error ↓↓↓

bayes error ↑

**Problem 6.** What is text normalization? What are the tradeoffs?

**Problem 7.** What is stop word elimination? What are the trade-offs of stop word elimination?

**Problem 8.** What is the TF-IDF transform? What are the tradeoffs?

**Problem 9.** What is the hashing trick? What are the trade-offs of using the hashing trick?
References:

1. Hashing trick tutorial: `https://booking.ai/dont-be-tricked-by-the-hashing-trick-192a6aae3087`
2. Zipf's law: `https://en.wikipedia.org/wiki/Zipf%27s_law`
3. Excellent research paper on the Johnson-Lindenstrauss lemma: `https://papers.nips.cc/paper/7784-fully-understanding-the-hashing-trick`

## 2 Time

**Problem 10.** The discretization transform.

**Problem 11.** The sin/cos transform.

# 3 Graph metadata

**Problem 12.** Friendship features.

**Problem 13.** How can pagerank be used in twitter classification?

# 4 Generic

**Problem 14.** What is the unit-normalization transform? What are the tradeoffs?

**Problem 15.** What is the clipping transform? What are the tradeoffs?

**Problem 16.** What is the log transform? What are the tradeoffs?

**Problem 17.** What is the whitening transform? What are the tradeoffs?