Data Engineering

Introduction to data engineering

↳ Gather data from different sources
↳ Process & combine
↳ Store clean data for use

Data Pipelines

↳ typically seperate from DS
↳ But, DS might need to know some in smaller companies.

ETL Pipelines

ETL → Extract Transform Load

CSV
JSON
API
→ database

→ Extract

↳ Practiced

→ Transform
  ↳ clean & combine data
  ↳ encodings
  ↳ remove duplicates
  ↳ Dummy variables / Normalize
  ↳ remove outliers
  ↳ etc.

→ Data formats
  ↳ CSV ← comma sep. values
    imported w/ pandas
  ↳ JSON ← key:value pair file
  ↳ XML ← the same
  ↳ API
  ↳ SQL ← language to communicate w/
    databases

→ While cleaning data you might need to combine data or parse it to the correct data type

→ you may also need to delete rows or create appropriate values where possible.

→ Handling outliers
  ↳ Cause, who knows? (Programmatic or human error)
  ↳ Statistical tests to remove
  ↳ Use pca or euclidian distance to check
  ↳ corr plot can even help

→ Scaling data
  ↳ Gives each feature a fair shot

→ Feature engineering
  ↳ Creating new features from old ones
    (helps w/ underfitting)
    It increases variance of model.

(Load)
  ← Store data appropriately

Recap.
→ Data engineering - prepare data
→ done w/ ETL

NLP pipelines

May not be linear

Text processing → Feature extraction → Modeling

Text processing: clean words

Feature extraction: features in

processing → example: website: remove HTML tags
→ creates plain text.

example library "Beautiful Soup"
↳ returns page as soup
↳ transforms to plain text w/ "get_text"
↳ possible to find tags w/ "find_all"

plain text!

Text processing
→ Normalization
↳ remove variations of the same words
↳ remove punctuation (?)

→ Tokenization
↳ take 1 string & split into multiple strings
(NLTK - used to perform some of this cleaning)

→ Stopword removal
↳ remove words that are not important to text analysis

→ Part of speech tagging
↳ Takes string & classifies based on being noun/adverb/etc.

→ Named entity recognition
↳ can recognise person/org./GPE/etc.

→ Stemming & lemmatization
↳ takes word & removes further variation

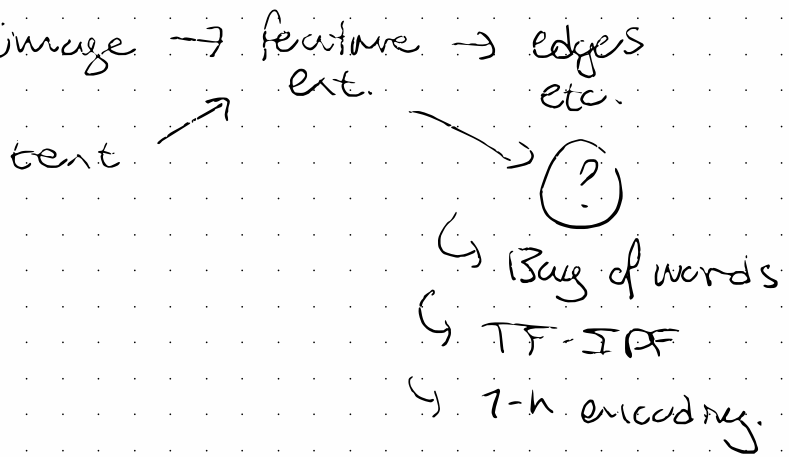
branch, ing, ed, es } branch

- Drawbacks of stemming
  - ↳ not complete words
  - ↳ may stem words incorrectly

Lemmatisation → uses a dict to find right root
↑ may not always be defined.

→ Feature extraction

↳ Text & pixels in an image have no inherent meaning



↳ Bag of words
↳ TF-IDF
↳ 1-h encoding.

13.0 W → document / word matrix
↳ dot prod of vectors is document or word similarity.

- TF-IDF
  - ↳ BOW is naive (doesn't attach importances to words)
  - ↳ add 1 row to BOW w/ Σ of word freq.
    - ↳ divide each # of buy number in this row for each col.

- one hot enc.
  - ↳ Similar to BOW only it's a word/word matrix ☹
  - ↳ Doesn't work well for large documents

- word embeddings
  - ↳ cluster words based on similarity in 3d space.

Other
word2vec
GloVe
↳ good for deep learning

→ Modeling
- ↳ use features to make predictions for seen & unseen data.
- ↳ create a model to predict on any target you want.

→ Machine Learning Pipelines

Automate ML workflows

example: classify corporate messages

Step 1: clean/tokenize
Step 2: fit a model
Step 3: wrap everything into a pipeline

↳ list of key, value pairs.

ie pipeline performs necessary steps before fitting/predicting

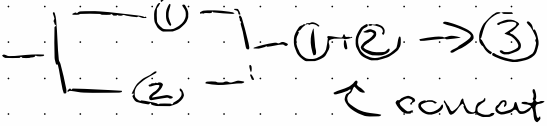
← All of these steps must be transformers (except final)

Adv of pipelines

- Simple
- Makes workflow easier to handle/maintain

Feature unions

↳ Allows for the creation of multiple features at once in a single pipeline

ie 

Custom transformers

↳ in the case that your pipeline requires complicated transformations, you can also make custom ones.

↳ create classes that extend sklearn.base's BaseEstimator & TransformerMixin

↳ need to create fit & transform methods

↳ or use sklearn.preprocessing's FunctionTransformer which turns a function into a transformer.

GridSearch

pseudo code

parameters = dict of params

```
svc = SVC()
clf = GS(svc, parameters)
clf.fit(X, y)
```

⚠ take care to transform data only after the split has taken place