4th) Applied Data Science - Data Engineering

by Kumaran Ponnambalam, Dedicated to Data Science Education Follow

This course focuses on the **Data Engineering**. It goes through the steps of **Data Acquisition, Cleansing, Transformation** and **Text Pre-processing**.

6 Lessons (1h 12m)

URL: https://www.skillshare.com/classes/Applied-Data-Science-4-Data-Engineering/840821259?via=similar-classes

1. About Applied Data Science Series 8:12

Course goal

- Train students to be full-fledged data science **practitioners** who could execute **end-to-end** data science projects to achieve **business results**
- The course is oriented towards existing software professionals
  - Heavily focused on programming and solution building
  - Limited, as-required exposure to math and statistics
  - Overview of ML concepts, with focus on using existing tools to develop solutions

Achievements

- Understand the concepts and life cycle of Data Science
- Develop proficiency to use R for all stages of analytics
- Learn Data Engineering tools and techniques
- Acquire knowledge of different machine learning techniques and know when and how to use them.
- Become a full-fledged Data Science Practitioner who can immediately contribute to real-life Data Science projects

Course structure

- Concepts of Data Science
- Data Science Life Cycle
- Statistics for Data Science
- R Programming
  - Examples
- Data Engineering
- Modeling and Predictive Analytics
  - Use cases
- Advanced Topics
- Resource Bundle

Data Engineering

- **Vital part and most difficult part of Data Science (painful, laborious, time consuming)**
- **Getting data from his source (making sure data is valid, reliable)**
- **Cleansing, transforming**
- **Putting them into repository**

2. Data Acquisition 16:01

Data Sources

Data sources

- Data Sources play a vital role in determining data and process architectures and workflows
  - Data Quality and Reliability
  - Network Planning
  - Fault tolerance
  - Security
  - Organizational boundaries

Enterprise data sources

- Typically RDBMS
- Within the organization's boundaries
- Easy accessibility and no rate limits
- Excellent Quality and reliability
- Data Guardians might be insecure of your use.
- Might need to work through organizational bureaucracy to get access

Cloud data sources

- Data hosted on the web like Sales Force, Marketo
- Access typically through SOAP or REST APIs
- Security is a pre-dominant factor
- Rate limits might apply
- Quality of Data would be excellent

Socila media data sources

- Facebook, twitter, LinkedIn, google
- Similar to Cloud Data Sources in most aspects
- Accessing public data about people and companies might involve privacy issues
- Rate limits are pretty restrictive

Web scraping as data source

- Scraping web sites is a raw but last way to get data that is otherwise not available
- Data is very dirty and would require a lot of cleaning
- Mostly text and require significant processing resources
- Security, privacy and intellectual property concerns

Data Formats

Data types of format

- Tables – from RDBMS
- CSV – most common data exchange format
- XML – configuration / meta data
- JSON – new age data exchange format
- Raw Text
- Binary – images, voice streams

Data Acquisition Trends

Acquisition intervals

- Types
  - Batch
  - Real time (push triggers)
  - Interval (e.g.: every 30 minutes)
  - Hybrid
- Determined by
  - Analytics needs
  - Availability
  - Rate limits
  - Reliability

3. Data Cleansing 10:50

Data Cleansing

Issue with Data Quality

- Invalid values
- formats
- Attribute dependencies
- Uniqueness
- Referential integrity
- Missing values
- Misspellings
- Misfielded values (value in the wrong field)
- Wrong references

Finding Data Quality Issues

- Sample Visual Inspection
- Automated validation code
  - Schema validation
- Outlier Analysis
- Exploratory Data Analysis

Fixing Data Quality Issues

- Fixing Data Quality issues is regular boilerplate coding in any language the team is comfortable with
- Fix the source if possible
- Find possible loopholes in data processing streams
- Analyze batches coming in and automate fixing
- Libraries and tools available

Data Imputation

- Any "value" present in a dataset is used by machine learning algorithms as valid values – including null, N/A, blank etc.
- This makes populating missing data a key step that might affect prediction results
- Techniques for populating missing data
  - Mean, median, mode
  - Multiple imputation
  - Use regression to find values based on other values
  - hybrid

4. Data Transformations 11:09

Data Transformations

Code samples

- Going forward, most examples will be covered as part of case studies

Data standardization

- Different sources of data follow different formats and hence standardization is required
- Having data in the same format and scale makes comparison and summarization activities easier

- Numbers
  - Decimal places
  - Log
- Date and Time
  - Time Zone
  - POSIX
  - Epoch
- Text Data
  - Name formatting (First Last vs Last, First)
  - Lower case/ Upper case/ Init Case

Binning

- Convert numeric to categorical data
- Pre-defined ranges are used to create bins and individual data records are classified based on this.
- New columns typically added to hold the binned data
- Binning is usually done when the continuous variable is used for classification.

Age	Age Range
35	20 - 40
23	20 - 40
11	01 - 20
65	60 - 80
40	40 - 60
51	40 - 60
20	20 - 40

Indicator variables

- Categorical variables are converted into Boolean data by creating indicator variables
- If the variable has n classes, then n-1 new variables are created to indicate the presence or absence of a value
- Each variable has 1/0 values
- The nth value is indicated by a 0 in all the indicator columns
- Indicator variables sometimes work better in predictions that their categorical counterparts

Pressure	Is High?	Is Medium?
High	1	0
Low	0	0
High	1	0
Medium	0	1
Medium	0	1
Low	0	0
High	1	0

Centering and Scaling

- Standardizes the range of values of a variables while maintaining their signal characteristics
- Makes comparison of two variables easier
- The values are "centered" by subtracting them from the mean value
- The values are "scaled" by dividing the above by the Standard Deviation.
- ML algorithms give far better results with standardized values

	Age	Height	Cent. Age	Cent. Height
	35	150	0.00	-1.66
	23	195	-0.74	1.92
	11	161	-1.47	-0.78
	65	165	1.84	-0.47
	40	180	0.31	0.73
	51	169	0.98	-0.15
	20	176	-0.92	0.41
Mean	35.00	170.86		
Std. Dev	16.30	12.56		

These are both forms of pre-processing numerical data*, that is, data consisting of numbers (as opposed to categories or strings).* Centering *a variable is subtracting the mean of the variable from each data point so that the new variable's mean is 0.* Scaling *a variable is multiplying each data point by a constant in order to alter the range of the data.*

All normalization means is scaling a dataset so that its minimum is 0 and its maximum 1 :

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Standardization is slightly different; it's job is to centre the data around 0 and to scale with respect to the standard deviation:

$$x_{standardized} = \frac{x - \mu}{\sigma}$$

where μ is the mean and σ is the standard deviation of the dataset. First note that these transformations merely change the range of the data and not the distribution. In French, it's called: centration-réduction (=> moyenne μ nulle, écart-type σ conservé).

Ref. https://www.datacamp.com/community/tutorials/preprocessing-in-data-science-part-1-centering-scaling-and-knn , http://w3.uohpsy.univ-tlse2.fr/UOHPSY/index.php?option=com_content&task=view&id=138&Itemid=30&limit=1&limitstart=2

5. Text Pre Processing TF IDF 14:53

Text Pre-Processing

Understanding how ML algorithms work

- ML Algorithms work with
  - numbers (continuous data)
  - classes (discrete/ categorical data)
- ML algorithms don't work with text.
- All textual data need to be converted into numbers or classes
- This is one of the main responsibilities of data pre-processing

Text cleansing

- Remove punctuation
- Remove white space
- Convert to lower case
- Remove numbers
- Remove stop words
- Stemming
- Remove other commonly used words

\+ **Lemmatization**

*With **stemming**, words are reduced to their word stems. A word stem need not be the same root as a dictionary-based morphological root, it just is an equal to or smaller form of the word - e.g. the words "cook," "cooking," and "cooked" are all to the same stem of "cook". This technic is not perfect- e.g. take the 4 words university, universal, universities, and universe, if a stemming algorithm that resolves these 4 words to the stem "univers" has overstemmed…*
*Stemming algorithms: Porter, Snowball, Lancaster. In Python, you should use NLTK (Natural Language Processing Tool Kit). **Lemmatization** is a more calculated process by resolving words to their dictionary form. A lemma of a word is its dictionary or canonical form! Lemmatizers require a lot more knowledge about the structure of a language (much more intensive process than heuristic stemming algorithm).*
Ref. https://towardsdatascience.com/stemming-lemmatization-what-ba782b7c0bd8

Term Frequency (TF) and IDF (Inverse Document Frequency)

- Text Documents are becoming inputs to ML more and more.
  - News items for classification
  - Email messages for spam detection
  - Text search
- Text need to be converted to equivalent numeric representation before ML can be used
- The most popular technique used is Term Frequency – Inverse Document Frequency (TF-IDF)
- TF-IDF output is table where rows represent documents and columns represent words
- Each cell provides a count / value that indicate the "strength" of the word with respect to the document

Ref. TF-IDF

TF-IDF formulae

Text Frequency (given a word w1 and Document d1)
= (# of times w1 occurs in d1) / (# of words in d1)

$$\mathrm{tf}_{1,1} = \frac{n_{1,1}}{\sum_k n_{k,1}}$$

Inverse Document Frequency (given a word w1)
= log e (Total # of docs / Total docs with w1)

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

où :

- $|D|$: nombre total de documents dans le corpus ;
- $|\{d_j : t_i \in d_j\}|$: nombre de documents où le terme t_i apparaît (c'est-à-dire $n_{i,j} \neq 0$).

TF-IDF = TF * IDF

$$\mathrm{tfidf_{i,j}} = \mathrm{tf_{i,j}} \cdot \mathrm{idf_i}$$

Voir video pour exemple (at 12')

6. R Examples for Data Engineering 11:14

Database acquisition

library(RMySQL), dbConnect(), dbSendQuery(), fetch(), dbHasCompleted(), dbClearResult(), dbDisconnect()

https://cran.r-project.org/web/packages/RMySQL/index.html

Download file from web

download.file()

https://stat.ethz.ch/R-manual/R-devel/library/utils/html/download.file.html

Read file (csv, pdf) and Text Mining (corpus)

library(tm), getReaders(), library(pdftools), read.csv(), readPDF(), str(), content(), cat(), aReader <- readPDF(engine="pdftools")
doc <- aReader(elem = list(uri = docfile), language = "fr", id = "pdf1")
class(doc), str(doc), class(doc$meta), class(doc$content), head(), inspect(), length(), dir()
VCorpus(VectorSource(doc$content)), tm_map() with stripWhitespace
content_transformer(tolower), removeNumbers, removePunctuation, stopwords(kind = "fr", stemDocument, stripWhitespace, dtm = DocumentTermMatrix(corpus), removeSparseTerms()
dataset = as.data.frame(as.matrix(dtm))
library(wordcloud), as.matrix(), sort(colSums(datasetCloud),decreasing=TRUE), names(v), data.frame(word=myNames,freq=v), wordcloud() …

https://www.rdocumentation.org/packages/tm/versions/0.7-7/topics/readPDF
https://www.rdocumentation.org/packages/tm/versions/0.7-7/topics/Corpus

Scraping web pages

library(RCurl), getURL(), cat(down_page)

Accessing via REST (JSON, Curl)

library(httr), library(jsonlite), library(httpuv), oauth_app("github", "…", secret="…")
git_token <- oauth2.0_token(oauth_endpoints("github"), gitapi)
GET("https://api.github.com/users/kumaranpm", config(token=git_token))
content(sample_request)$blog
library(curl), curl_fetch_memory(), str(req), class(req), parse_headers(req$headers)
jsonlite::prettify(rawToChar(req$content)), jsonlite::parse_json(json_content), content[32]$user_search_url

Data cleansing

```
#finding outliers
```

c(-1, 3,4,12,6,8,4,5,7), quantile(), boxplot(), students_age[students_age < 0]

Data transformation

Dataset: ***mtcars*** *, str(car_data) ,* *summary*(car_data)

```
# factor conversion
car_data$fact_cyl <- as.factor(car_data$cyl)

# binning
quantile(car_data$hp)
car_data$bin_hp <- cut(car_data$hp, c(0,100, 200,300,400))

# indicator variables
car_data$is_4cyl <- ifelse( car_data$cyl == 4, 1, 0)
car_data$is_6cyl <- ifelse( car_data$cyl == 6, 1, 0)

# centering and scaling
car_data$scaled_mpg <- scale(car_data$mpg)[,1]
str(car_data)
```

	fact_cyl	bin_hp	is_4cyl	is_6cyl	scaled_mpg
Mazda RX4	6	(100,200]	0	1	0.15088482
Mazda RX4 Wag	6	(100,200]	0	1	0.15088482
Datsun 710	4	(0,100]	1	0	0.44954345
Hornet 4 Drive	6	(100,200]	0	1	0.21725341