PA164 Natural Language Learning

Lecture 06: AutoML for NLP

Vít Nováček

Faculty of Informatics, Masaryk University

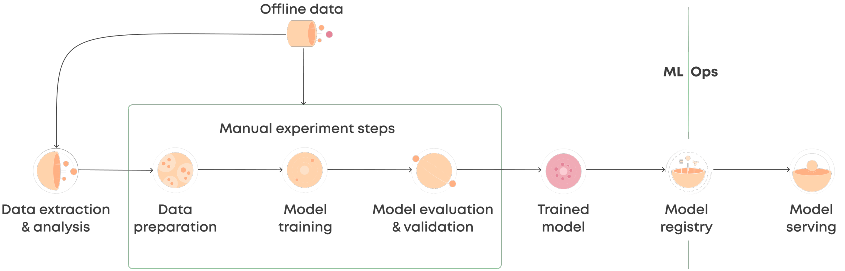
Autumn, 2024

MUNI

Outline

Outline

Typical ML Pipeline



[1] The image source: `https://valohai.com/machine-learning-pipeline/`. License: unknown.

The practical challenges of putting together a ML pipeline

- Data issues
  - Technical constraints: noise, incompleteness, inconsistency, sparsity, lack of (good) labels, feature engineering conundrums, ...
  - Ethical constraints: skewed representation of the problem domain, data sensitivity, ...
- Model issues
  - What model is best-suited to the problem?
  - What hyper-parameters will make the model work best?
  - How to explain the model predictions?
- Validation issues
  - What validation metric(s) to use?
  - How to interpret the results (including fair model comparison)?
- Deployment issues
  - How to ensure reproducibility and transparency?
  - Debugging and adaptation to new data "in the wild"

AutoML to the rescue!

- Many (it not most) of the ML pipeline construction steps are
  - Knowledge- and labour-intensive
  - Time-consuming, tedious and error-prone even if one can put in the qualified work
- Some of the steps can be automated, though
- Relatively straightforward
  - Model selection, hyper-parameter optimisation, deployment
- Not so straightforward
  - Data preparation, validation

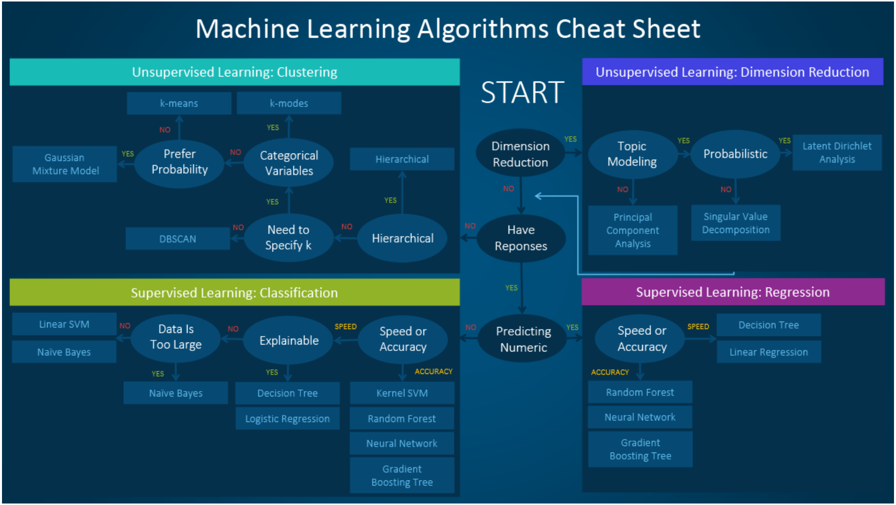
Outline

Automating the data preprocessing stage

- No generally applicable rules of thumb
- There are tools for some tasks, such as:
  - Feature engineering
    - Various feature selection and feature extraction techniques
  - Dimensionality reduction
    - SVD, PCA, Latent discriminant analysis, ...
- Noise or bias in the data, however, have to be tackled more or less manually
  - Exploratory and statistical analysis first, ad hoc solutions then

Outline

Model selection – sample guidelines



[2] The image source: `https://easyblog729.netlify.app/sklearn-machine-learning-cheat-sheet.html`. License: unknown.

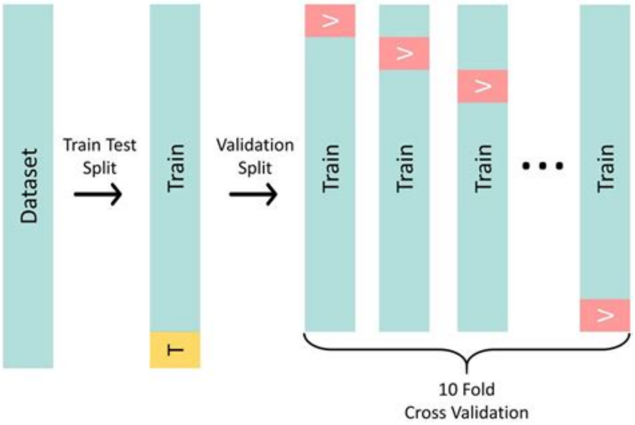
Automating the model selection

- Either empirically...
  - Running various applicable algorithms iteratively on limited data sample(s)
  - Comparing their performance using a specific endpoint (e.g., accuracy, runtime or both)
  - Selecting the best-performing one
  - Examples of existing techniques:
    - Average ranking, active testing, ...
- ...or making the model choice another hyper-parameter

Model parameters vs. hyper-parameters

- Parameters
  - Optimised during the training
  - Examples:
    - ★ The vector **w** in a linear regression model $y = \mathbf{w}^T\mathbf{x}$
    - ★ The weights of a neural network
    - ★ The splits and terminal nodes for a decision tree
- Hyper-parameters
  - Must be specified before the training
  - Examples:
    - ★ The kernel of an SVM model
    - ★ The number and type of layers, activation functions, etc., in neural networks
    - ★ The depth and branching factor of a decision tree

Hyper-parameter optimisation via cross-validation

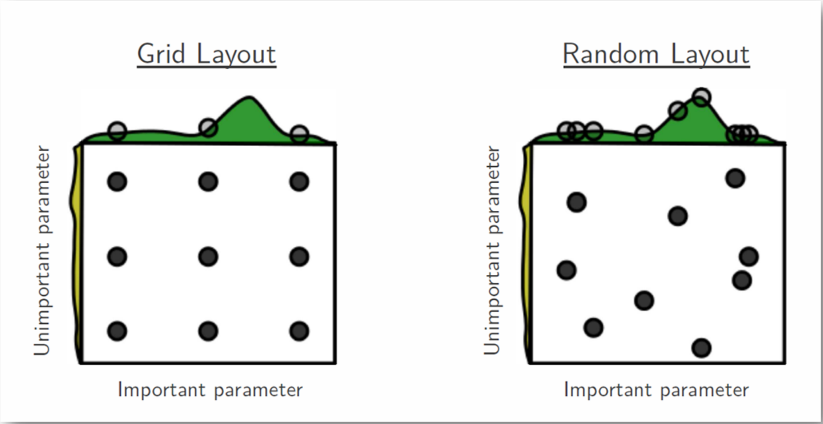


[3] The image source: Silveira Kupssinskü, Lucas, et al. "A method for chlorophyll-a and suspended solids prediction through remote sensing and machine learning." Sensors 20.7 (2020): 2125. License: CC BY 3.0.

Hyper-parameter optimisation techniques

- Random search
  - Random sampling of combinations from possible hyper-parameter ranges
  - Not exhaustive (i.e., can miss some optimum), but less computationally expensive
- Grid search
  - Exhaustive search over all combinations from possible hyper-parameter ranges
  - Can be rather slow for many hyper-parameters or their large ranges
  - May miss optima in higher-dimensional searches
- Examples of other widely-used techniques
  - Bayesian optimisation
  - Evolutionary search

Hyper-parameter optimisation example – grid vs. random search



[4] The image source: Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." Journal of machine learning research 13.2 (2012). License: unknown.

Examples of hyper-parameter optimisation frameworks

- *scikit-learn*
  - Various common techniques implemented (grid search, random search, successive halving, ...)
- Optuna
  - A tool dedicated solely to (hyper-parameter) optimisation
  - SoA algorithms, easy parallelisation
- SMAC
  - Another dedicated tool developed by an academic AutoML group in Germany

Neural architecture search

- Automating the design of neural networks
- Three main components:
  - The search space – the type(s) of network to be designed and optimized
  - The search strategy – the approach used to explore the search space
  - The performance estimation – evaluating the network performance from its design (i.e., without actually constructing and training it)
- Examples of applicable techniques:
  - Reinforcement learning, evolutionary computing, Bayesian optimisation, hill-climbing, multi-objective search, . . .

Notes on meta-learning

- Subfield of machine learning closely related to AutoML
- ML algorithms applied to metadata about ML experiments
- A definition of a meta-learning system:
    - The system must include a learning subsystem.
    - Experience is gained by exploiting meta-knowledge extracted in a previous learning episode on a single dataset, or from different domains.
    - Learning bias (i.e., what kind of ML hypotheses are tested) must be chosen dynamically.

Outline

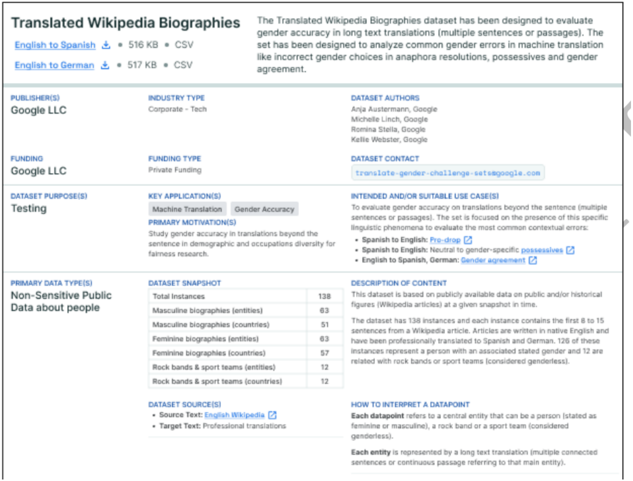
MLOps

- Definition:
  - A set of tools and techniques...
  - used for deploying and maintaining ML models...
  - in production...
  - reliably and efficiently.
- Uses of MLOps:
  - Deployment and automation, reproducibility, diagnostics, scalability
  - Governance and regulatory compliance
  - Collaboration, business uses
  - Monitoring and management

Data and model cards

- An emerging semi-standardised way of representing and sharing meta-data about ML data sets and models
- Crucial for transparency, reproducibility and accountability of ML experiments
- Encouraging and supporting informed decision making for end users
- Ideally, following the FAIR principles:
  - Findable
  - Accessible
  - Interoperable
  - Reusable

Data card example



Translated Wikipedia Biographies

English to Spanish • 516 KB • CSV

English to German • 517 KB • CSV

The Translated Wikipedia Biographies dataset has been designed to evaluate gender accuracy in long text translations (multiple sentences or passages). The set has been designed to analyze common gender errors in machine translation like incorrect gender choices in anaphora resolutions, possessives and gender agreement.

PUBLISHER(S)
Google LLC

INDUSTRY TYPE
Corporate - Tech

DATASET AUTHORS
Anja Austermann, Google
Michelle Linch, Google
Romina Stella, Google
Kellie Webster, Google

FUNDING
Google LLC

FUNDING TYPE
Private Funding

DATASET CONTACT
translate-gender-challenge-sets@google.com

DATASET PURPOSE(S)
Testing

KEY APPLICATION(S)
Machine Translation · Gender Accuracy

PRIMARY MOTIVATION(S)
Study gender accuracy in translations beyond the sentence in demographic and occupations diversity for fairness research.

INTENDED AND/OR SUITABLE USE CASE(S)
To evaluate gender accuracy on translations beyond the sentence (multiple sentences or passages). The set is focused on the presence of this specific linguistic phenomena to evaluate the most common contextual errors:

- **Spanish to English:** Pro-drop
- **Spanish to English:** Neutral to gender-specific possessives
- **English to Spanish, German:** Gender agreement

PRIMARY DATA TYPE(S)
Non-Sensitive Public Data about people

DATASET SNAPSHOT

Total Instances	138
Masculine biographies (entities)	63
Masculine biographies (countries)	51
Feminine biographies (entities)	63
Feminine biographies (countries)	57
Rock bands & sport teams (entities)	12
Rock bands & sport teams (countries)	12

DESCRIPTION OF CONTENT
This dataset is based on publicly available data on public and/or historical figures (Wikipedia articles) at a given snapshot in time.

The dataset has 138 instances and each instance contains the first 8 to 15 sentences from a Wikipedia article. Articles are written in native English and have been professionally translated to Spanish and German. 126 of these instances represent a person with an associated stated gender and 12 are related with rock bands or sport teams (considered genderless).

DATASET SOURCE(S)

- **Source Text:** English Wikipedia
- **Target Text:** Professional translations

HOW TO INTERPRET A DATAPOINT
Each datapoint refers to a central entity that can be a person (stated as feminine or masculine), a rock band or a sport team (considered genderless).

Each entity is represented by a long text translation (multiple connected sentences or continuous passage referring to that main entity).

[5] The image source: Pushkarna, Mahima, Andrew Zaldivar, and Oddur Kjartansson. "Data cards: Purposeful and transparent dataset documentation for responsible ai." Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency. 2022. License: unknown.

Model card example

Model Details

- Developed by researchers at Google and the University of Toronto, 2018, v1.
- Convolutional Neural Net.
- Pretrained for face recognition then fine-tuned with cross-entropy loss for binary smiling classification.

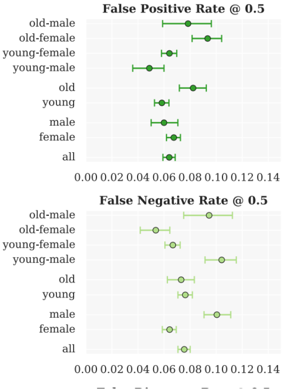
Intended Use

- Intended to be used for fun applications, such as creating cartoon smiles on real images; augmentative applications, such as providing details for people who are blind; or assisting applications such as automatically finding smiling photos.
- Particularly intended for younger audiences.
- Not suitable for emotion detection or determining affect; smiles were annotated based on physical appearance, and not underlying emotions.

Factors

- Based on known problems with computer vision face technology, potential relevant factors include groups for gender, age, race, and Fitzpatrick skin type; hardware factors of camera type and lens type; and environmental factors of lighting and humidity.
- Evaluation factors are gender and age group, as annotated in the publicly available dataset CelebA [36]. Further possible factors not currently available in a public smiling dataset. Gender and age determined by third-party annotators based on visual presentation, following a set of examples of male/female gender and young/old age. Further details available in [36].

Quantitative Analyses



[6] The image source: Mitchell, Margaret, et al. "Model cards for model reporting." Proceedings of the conference on fairness, accountability, and transparency. 2019. License: unknown.

Outline

Useful References – hyper-parameter optimisation

- Bergstra, James, et al. "Algorithms for hyper-parameter optimization." Advances in neural information processing systems 24 (2011).
- Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." Journal of machine learning research 13.2 (2012).
- Yang, Li, and Abdallah Shami. "On hyperparameter optimization of machine learning algorithms: Theory and practice." Neurocomputing 415 (2020): 295-316.

Useful References – meta-learning, NAS and AutoML in general

- Vilalta, Ricardo, and Youssef Drissi. "A perspective view and survey of meta-learning." Artificial intelligence review 18 (2002): 77-95.
- Vanschoren, Joaquin. "Meta-learning: A survey." arXiv preprint arXiv:1810.03548 (2018).
- Ren, Pengzhen, et al. "A comprehensive survey of neural architecture search: Challenges and solutions." ACM Computing Surveys (CSUR) 54.4 (2021): 1-34.
- Pushkarna, Mahima, Andrew Zaldivar, and Oddur Kjartansson. "Data cards: Purposeful and transparent dataset documentation for responsible ai." Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency. 2022.
- Mitchell, Margaret, et al. "Model cards for model reporting." Proceedings of the conference on fairness, accountability, and transparency. 2019.
- Barbudo, Rafael, Sebastián Ventura, and José Raúl Romero. "Eight years of AutoML: categorisation, review and trends." Knowledge and Information Systems (2023): 1-53.

Useful References – selected online resources

- https://www.automl.org/
- https://scikit-learn.org/stable/modules/grid_search.html
- https://www.cs.jhu.edu/~kevinduh/a/automl-tutorial-2023/2305-EACL-AutoMLtutorial.pdf