CMT307: Applied ML Session 2

Data preprocessing, feature engineering/selection/extraction

Reminders (last session)

- Basic Machine Learning **introduction**.
- Set up **Python 3** + **libraries** (numpy, nltk, etc.) + **Jupyter Notebooks** (Google Colab or local).
- Refreshers of programming and mathematics (**online tutorials**).
- Python notebook on basic **text preprocessing** with *nltk* and **vector manipulation** with *numpy* (Solutions to the exercises now available in learning central).

Outline

- ➢ Research/Practice opportunities
- ➢ Machine learning pipeline
- ➢ Classification vs. Regression
- ➢ Feature engineering
- ➢ Feature selection
- ➢ Hands on!

Research/Practice Opportunities

My research interests

Natural Language Processing (NLP)!

NLP is a subfield of AI that studies how to program computers to analyze and **understand** natural language data.

Natural Language Processing (NLP)

Some topics:

- ➢ Language understanding (**semantics**)
- ➢ **Multilinguality** and cross-lingual transfer
- ➢ Application of NLP in **social media**.
- ➢ Vector Space Models: word/relation/contextualized **embeddings**

Come talk to me if you are interested in any of these topics or would like to write your master thesis on NLP!

Kaggle and SemEval

- **Kaggle** (https://www.kaggle.com/): Many datasets and competitions on data science (most related to machine learning).

- **SemEval** (http://alt.qcri.org/semeval2020/): Annual research competitions on NLP tasks. Most of them framed as a machine learning problem (training and test sets provided). 12 tasks (potential MSc dissertation topics), deadline January 2020.

Opportunities in Cardiff

Fully-funded PhD studentship on “Analysing treatment resistance in psychiatric disorders through large-scale electronic clinical records”.

Machine Learning and Data Science. Experience in biomedicine NOT required.

Supervisors from both School of Medicine and Computer Science.

Application deadline: November 25. **Start date**: October 2020.

[Only for EU/UK students]

Activities in Cardiff

- **AI Wales** (https://www.meetup.com/AI-Wales/): Monthly meetings about AI (machine learning, NLP, computer vision, etc.).

- **Pydata Cardiff** (https://www.meetup.com/PyData-Cardiff-Meetup/): Data analysis community around Python.

Both are free, including workshops, technical talks and refreshments.

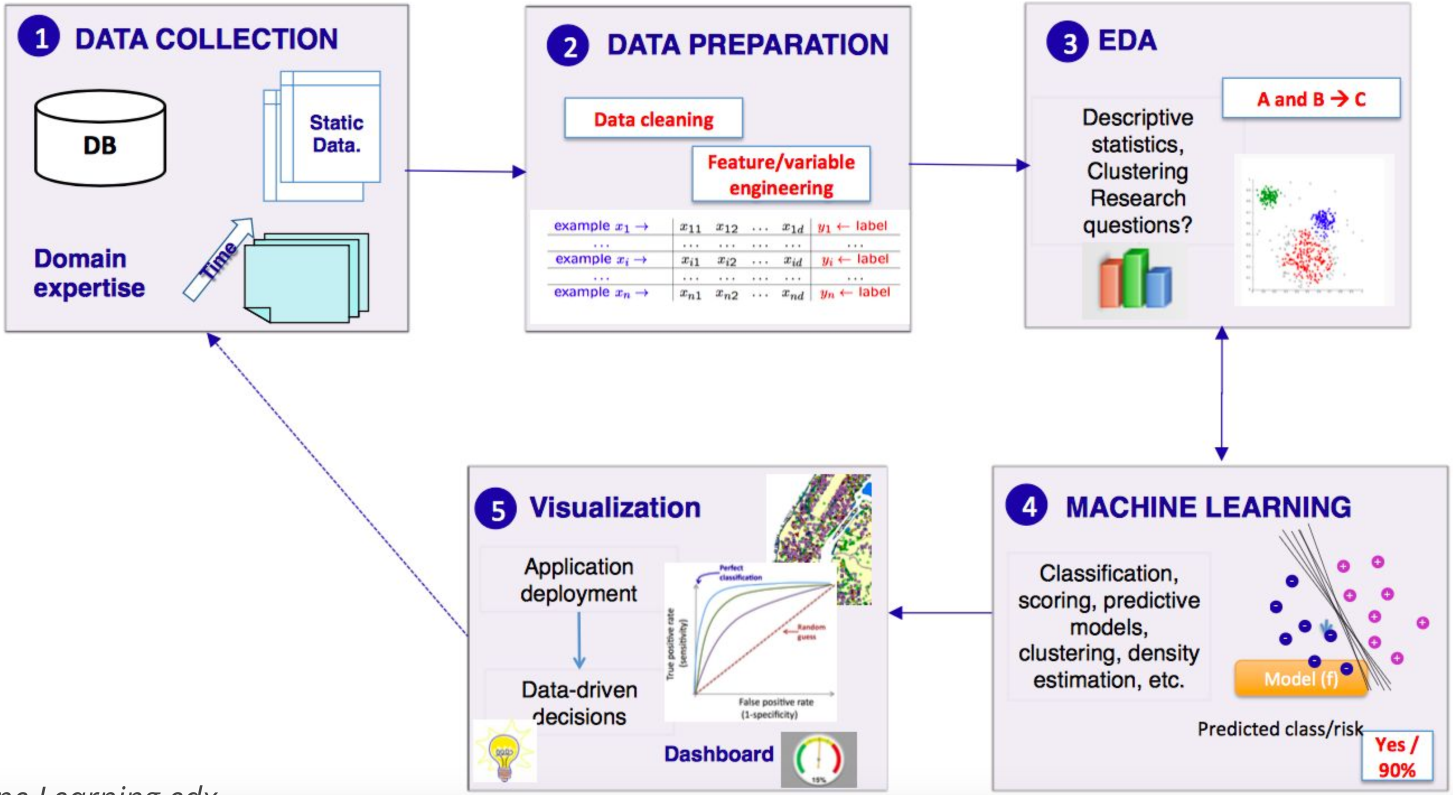
Machine Learning Pipeline

Machine Learning pipeline

Machine learning generally involves several stages, from **data collection** and **preprocessing**, to **training** and **analysis**.

All stages are important, and we should be careful and **understand all key stages** to be a successful machine learning practitioner.

Machine Learning pipeline



Slide credit: *Machine Learning edx*

Machine Learning pipeline: training stage

Formally, given a number *n* of training examples (x_1, y_1), …, (x_N, y_N) where x_i represents an **input** (or *feature*) vector and y_i an **output** label, we need to find a **function** f: X -> Y, where X represents the input space and Y the output space.

In this module we are NOT going to learn how to learn that function *f*, but rather how to use existing ones, and all remaining stages of the pipeline.

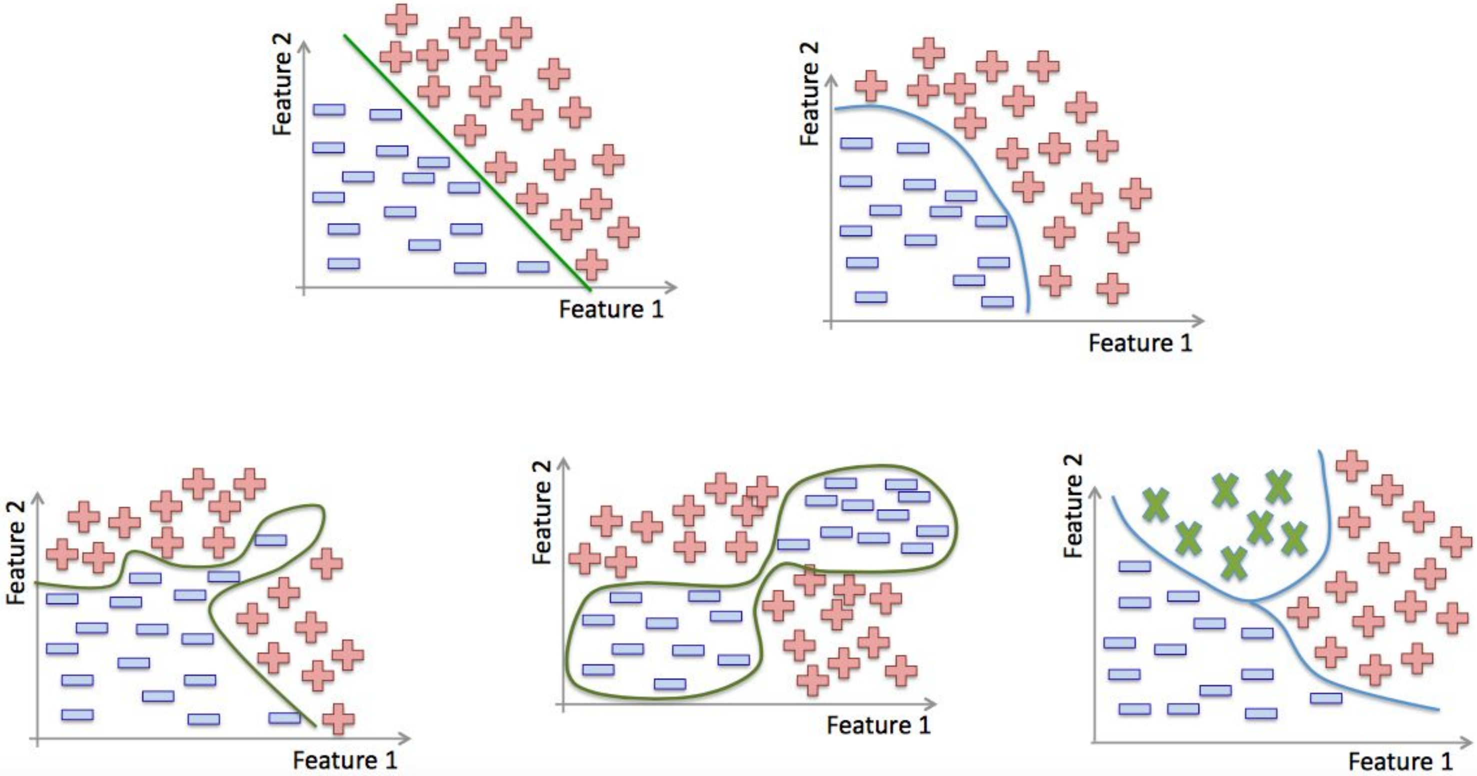
Classification vs. Regression

Classification vs. Regression

- ➢ We refer to classification problems to those where output variables are **categories** (e.g. “positive” or “negative”, “spam” or “not spam”).
- ➢ When the output variable is a continuous value (e.g. a real number), we refer to it as **regression**.

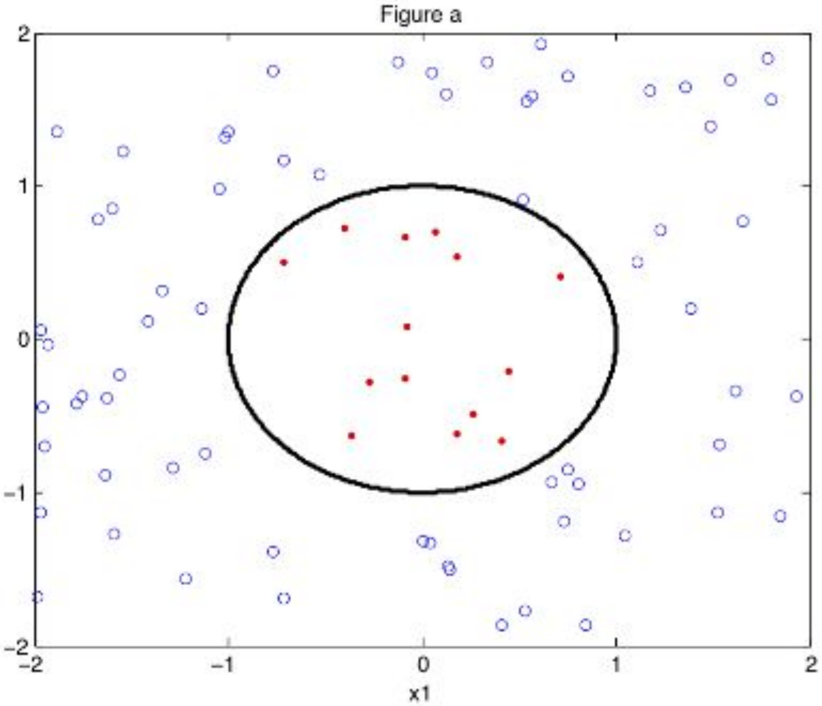
Today we are going to mostly focus on classification.

Supervised learning: Classification



Slide credit: *Machine Learning edx*

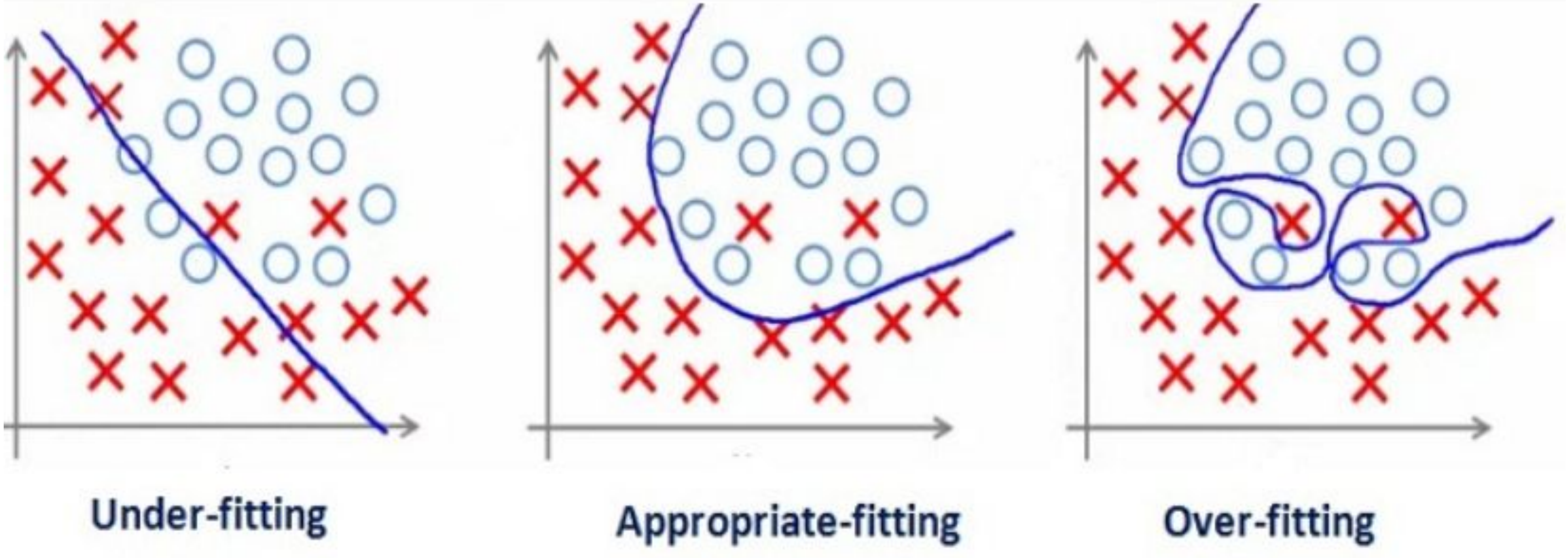
Supervised learning: Non-linear classification



Neural networks can help us solve non-linear problems (2nd Semester!)

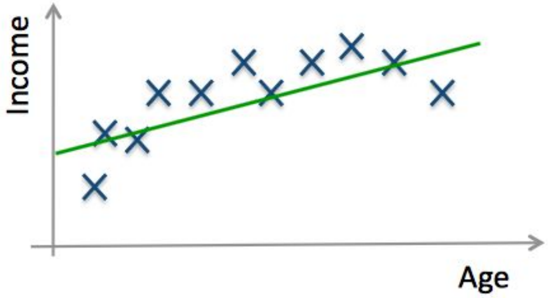
Slide credit: *Machine Learning edx*

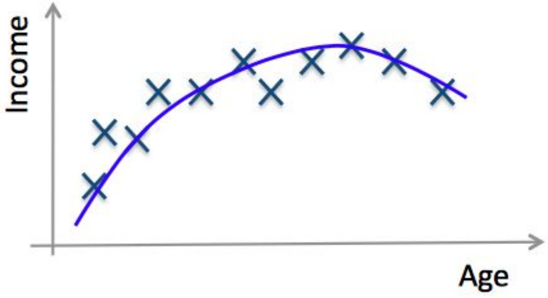
Overfitting vs. underfitting (classification)

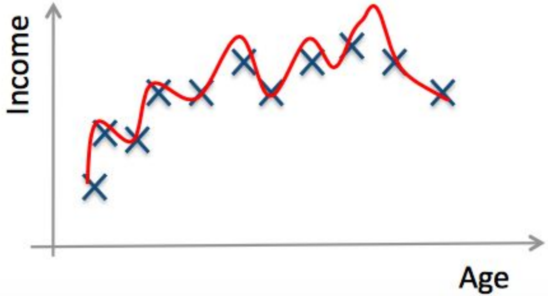


Slide credit: *GreyAtom*

Overfitting vs. underfitting (regression)







Slide credit: *Machine Learning edx*

Feature Engineering

Feature engineering

Usually, data cannot be easily fed into Machine Learning algorithms.

We need to **transform data into vectors** and this is often not trivial.

Using our **knowledge about the data domain**, we can come up with *“feature vectors”* which can be extracted from the data. These features can then be fed directly into any Machine Learning algorithm.

Feature engineering (Sentiment analysis)

- I liked the movie 😀
- The movie was awesome 😀
- It was quite boring ☹️
- I enjoyed the movie 😀
- It was great! 😀
- The main actor was terrible ☹️

Raw data

Feature engineering (Sentiment analysis)

- I liked the movie
- The movie was awesome
- It was quite boring
- I enjoyed the movie
- It was great!
- The main actor was terrible

[0.22, 1.52, ... , -1.44, 0.11]

[-1.33, 5.62, ... , -1.23, -9.22]

[0.88, 2.83, ... , 4.43, 0.89]

[11.23, 8.52, ... , -1.23, 6.33]

[-1.66, -1.33, ... , 8.23, 0.22]

[0.31, -6.51, ... , -7.63, 3.65]

Raw data **Vectors**

How to choose features? Many ways

- I liked the movie
- The movie was awesome
- It was quite boring
- I enjoyed the movie
- It was great!
- The main actor was terrible

How to choose features? Many ways

- ➢ **I liked the movie**
- ➢ **The movie was awesome**
- ➢ **It was quite boring**
- ➢ **I enjoyed the movie**
- ➢ **It was great!**
- ➢ **The main actor was terrible**

Frequency of the words

How to choose features? Many ways

- I liked the movie
- The movie was **awesome**
- It was quite **boring**
- I enjoyed the movie
- It was **great**!
- The main actor was **terrible**

Frequency of the words

Frequency of adjectives only

How to choose features? Many ways

- I **liked** the movie
- The movie was **awesome**
- It was quite **boring**
- I **enjoyed** the movie
- It was **great**!
- The main actor was **terrible**

Frequency of the words

Frequency of **adjectives** only

Frequency of adjectives + verbs

How to choose features? Many ways

- I **liked** the movie
- The movie was **awesome**
- It was quite **boring**
- I **enjoyed** the movie
- It was **great**!
- The main actor was **terrible**

Frequency of the words

Frequency of adjectives only

Frequency of adjectives + verbs

Count positive and negative words

How to choose features? Many ways

- **[I liked]** the movie
- The **[movie was]** awesome
- It was **[quite boring]**
- **[I enjoyed]** the movie
- It was **[great !]**
- The **[main actor]** was terrible

Frequency of the words

Frequency of adjectives only

Frequency of adjectives + verbs

Count positive and negative words

Bigrams (or n-grams)

How to choose features? Many ways

- ➢ I liked the movie
- ➢ The movie was awesome
- ➢ It was quite boring
- ➢ I enjoyed the movie
- ➢ It was great!
- ➢ The main actor was terrible

Frequency of the words

Frequency of adjectives only

Frequency of adjectives + verbs

Count positive and negative words

Bigrams (or n-grams)

....

Feature Selection

Feature selection

Feature selection consists of selecting a **subset of relevant features** from the full feature set.

Why feature selection?

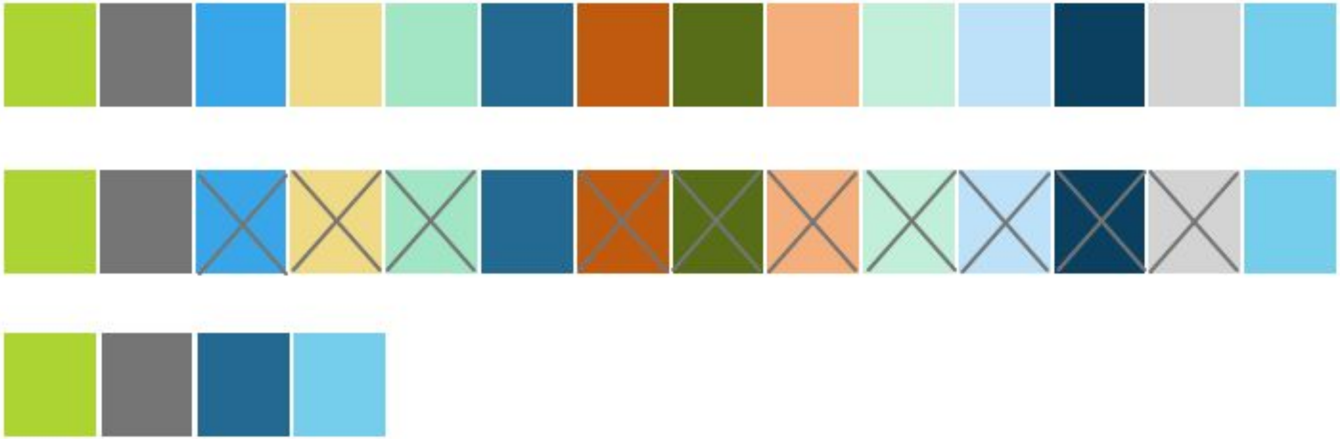
- ➢ **Simplify** models
- ➢ **Less time** required to train and predict
- ➢ Avoid sparsity or the ***curse of dimensionality***
- ➢ Reduce **overfitting**

Feature selection

Full feature set

Identify useful features

Selected feature set



Slide credit: *Mehul Ved*

Feature selection: Methods

Can be divided into:

- **Unsupervised**: Make use of unlabelled data only (e.g. remove sparse or low-variance features, based on their entropy, etc.).

- **Supervised:** Make use of the output labels, and generally are aimed at **removing features that are not relevant** or do not help to improve the performance of the machine learning model.

Supervised Feature Selection Methods

Supervised feature selection methods can be further split into:

- **Filter methods**: Statistical tests to score each feature.
  - *Examples: Chi-squared test, correlation.*
- **Embedded**: Learn the most relevant features while the model is being created. Regularization is the most common technique.
  - *Examples: SMLR, LASSO, Ridge Regression*
- **Wrapper**: Consider the selection of features as a search problem.
  - *Examples: Forward/Backward selection, Recursive Feature Elimination*

Feature extraction

Feature extraction is similar to feature selection in which it **reduces the number of features** from the original feature set.

However, in feature extraction, new **features are created**, unlike in feature selection where a subset of existing features is selected.

Common methods: PCA, LDA, Autoencoders, etc.

More information on feature selection and extraction methods:
https://elitedatascience.com/dimensionality-reduction-algorithms

School's private Stack Overflow (reminder)

https://stackoverflow.com/c/comsc

Post your questions related to the course here!

Add the tags ***cmt307*** and ***machine-learning*** to your question.

Hands on!

Python notebook with exercises
about **data preprocessing** and **feature selection** available at Learning Central