Data Mining 2

Topic 01 : Module Introduction

Lecture 01 : Module Overview

Dr Kieran Murphy

Department of Computing and Mathematics, Waterford Institute of Technology.
(Kieran.Murphy@setu.ie)

Spring Semester, 2026

Outline

- Module motivation and aims.
- The three components of a Machine Learning Problem
- Data mining / Machine Learning workflow

Outline

What is Data Mining ?

We are drowning in data but starving for knowledge!

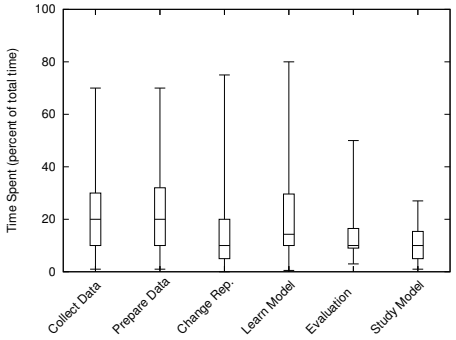
Necessity is the mother of invention $\Rightarrow$ Data Mining $\approx$ Automated analysis of massive data sets.

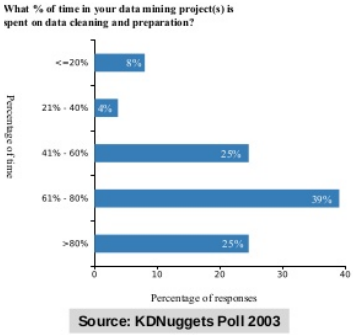
Definition 1 (Data Mining)

The non-trivial extraction of implicit, previously unknown and potentially useful knowledge from data in large data repositories

non trivial	— obvious knowledge is not useful (we already know it)
implicit	— hidden difficult to observe knowledge
previous unknown	— if known then, why go to this effort?
potentially useful	— actionable easy to understand

Data Mining (Model Building) is less than half of Data Mining

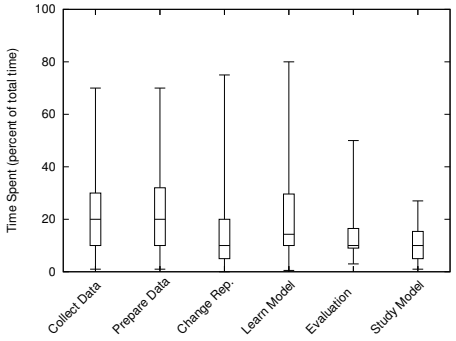


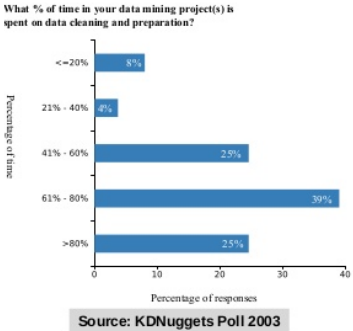


- Boxplots: median is 20% on collecting data, 20% on preparing data, and 10% on changing data representation — all before starting on model.
- Bar chart — data cleaning and preparation consumes at least 80% of project time for 25% of the participants, and 61% to 80% for another 39%.

See Study on the Importance of and Time Spent on Different Modeling Steps, 2012

Data Mining (Model Building) is less than half of Data Mining



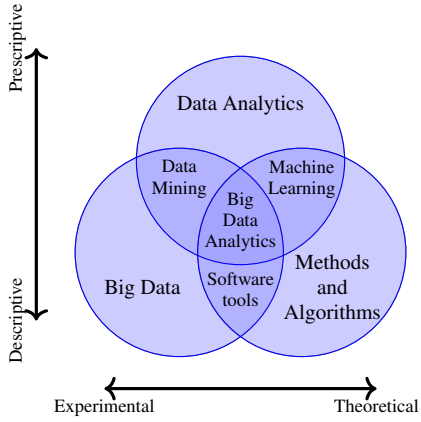


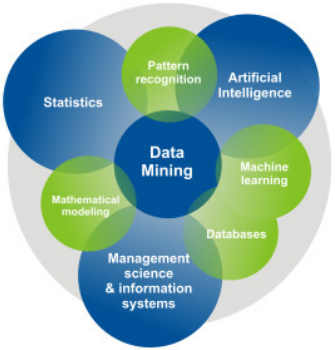
- Boxplots: median is 20% on collecting data, 20% on preparing data, and 10% on changing data representation — all before starting on model.
- Bar chart — data cleaning and preparation consumes at least 80% of project time for 25% of the participants, and 61% to 80% for another 39%.

See Study on the Importance of and Time Spent on Different Modeling Steps, 2012

Related Disciplines — Data Mining vs Data Analytics vs Data Science†

- Data Mining is about finding the patterns in a data set, and using these patterns to make predictions.
- Data Science is a field of study which includes everything from Big Data Analytics, Data Mining, Predictive Modelling, Data Visualisation, Mathematics, and Statistics.





*In other words, have we titled this module correctly? Probably not, and it should be called Data Analytics 2 or Data Science 2

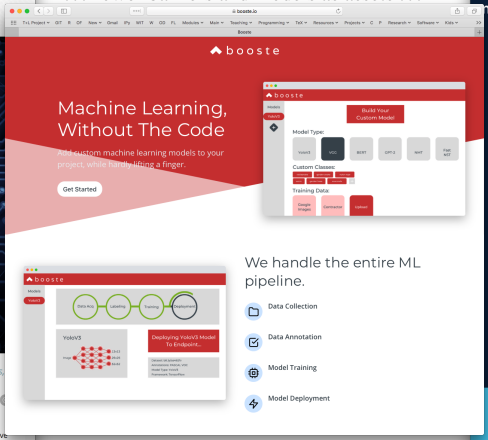
Data Science Mind Map

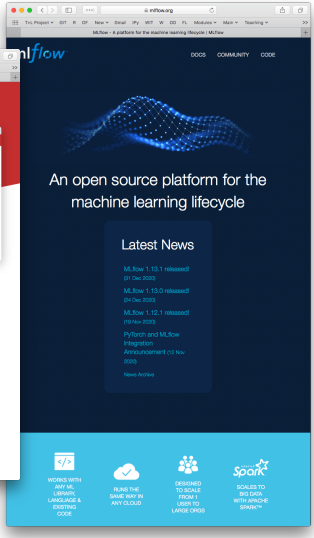
Data Science in 2021 — ML Models as assets, ML Deployment Services

Not all are believers…

… lower barriers and models as assets …

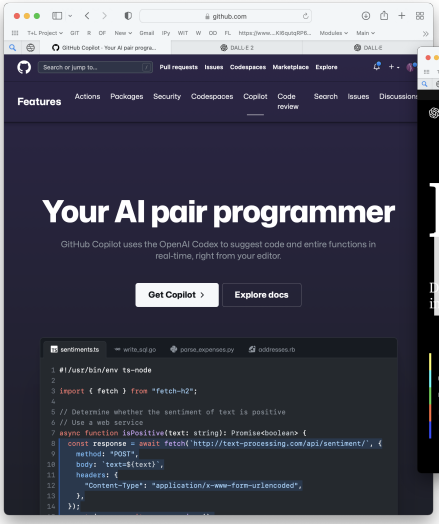
… MLOps



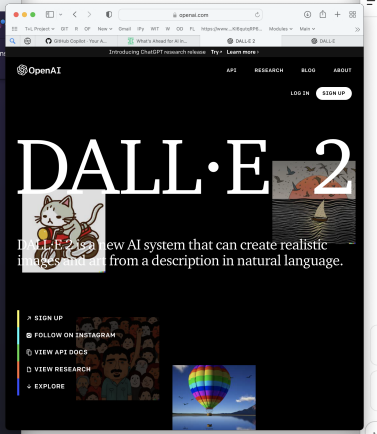


Data Science in 2022 — Generative AI and LLM

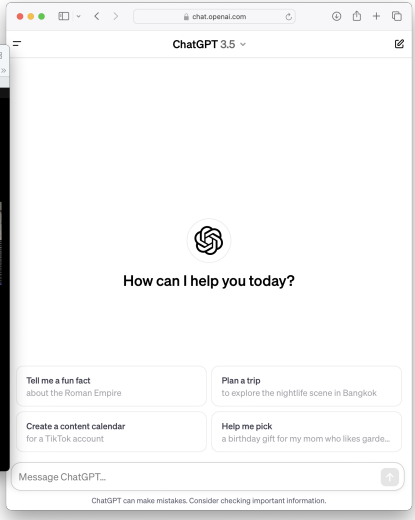
Generating code …



… images from text …

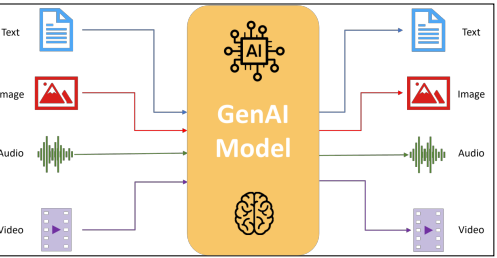


… any text (using chatGPT)



Data Science in 2023 — Generative AI going Mainstream

Multimodel models



Open sourcing of models

Sharing and reusing of models (Hugging Face)

General model classes

PreTrainedConfig
- is_encoder_decoder
- output_attentions
- from_pretrained()
- save_pretrained()

Each model specific config inherits all generic attributes and functions from PreTrainedConfig and adds model specific attributes.

reference

PreTrainedModel
- config
- from_pretrained()
- save_pretrained()
- init_weights()
- generate()

Each model specific PreTrainedModel has access to self.config, inherits all functions from PreTrainedModel and provides a model specific initialization method used by init_weights()

PreTrainedConfig.from_pretrained(dir):
Load serialized configuration object from dir (locally or from url). File is expected to be found as dir/config.json.

PreTrainedConfig.save_pretrained(dir): Serializes configuration instance as dir/config.json.

PreTrainedModel.from_pretrained(dir): 1st calls init_weights() to initialize all weights. Then loads serialized model from dir (locally or from url) and overwrites all initialized weights found in ed values. Also calls ed(dir) and sets to self.config.

ained(dir): Serializes model bin.

Generates output_ids given lel.forward() in a while loop

Quantization

Floating point 3452.3194 → Integer 3452

32 bit → 8 bit

Specific to each model

BrandNewBertConfig
inherit from **PreTrainedConfig**
- vocab_size
- num_hidden_layers
- num_attention_heads

BrandNewBertConfig is the model specific configuration and contains all parameter that are required to instantiate a BrandNewBertModel, such as *num_hidden_layers*, etc. Every parameter should have a default value that corresponds to the main model checkpoint to be added

BrandNewBertPreTrainedModel
inherit from **PreTrainedModel**
- base_class_prefix = "brand_new_bert"
- _init_weights()

BrandNewBertPreTrainedModel is the model specific PreTrainedModel class. It is important to define the initialization strategy and common class attributes, such as *base_class_prefix* for all BrandNewBert implementations.

Both the "base" model without any head and models with a specific head inherit from a model specific PreTrainedModel

BrandNewBertModel
inherit from **BrandNewBertPretrainedModel**
- brand_new_bert_encoder
- forward()

reference

BrandNewBertForMaskedLM
inherit from **BrandNewBertPretrainedModel**
- brand_new_bert
- masked_lm_head
- forward()

BrandNewBertModel is the "base" model that has torch.nn.Modules, such as *brand_new_bert_encoder* as attributes and outputs just the encoded hidden states.

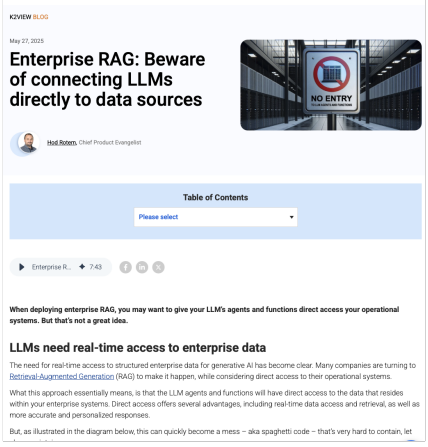
BrandNewBertModel.forward(): defines the inference function for models. It is called when running *model(input_ids)*. The goal for every new model should be that **BrandNewBertModel.forward()** is a self-contained function of the model file so that the reader does not have to look into other files to under it

BrandNewBertForMaskedLM is a head-specifc model that has **BrandNewBertModel** and a specific head layer, called *masked_lm_head* as attributes.

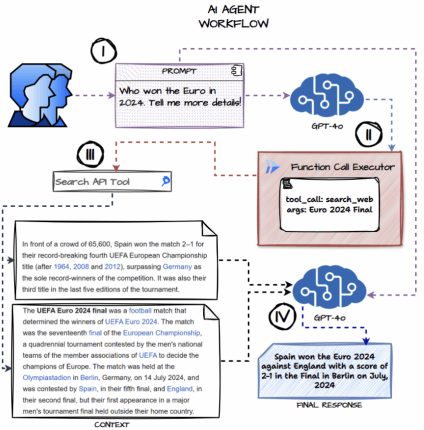
BrandNewBertForMaskedLM.forward(): calls *brand_new_bert.forward()* and *masked_lm_head.__call__()* to generated the head specific outputs, e.g. the logits in this case. The function is usually very short

Data Science in 2024 — AI Agents and Enterprise RAG

Retrieval-Augmented Generation (RAG): Grounding LLMs in private enterprise data to reduce hallucinations.

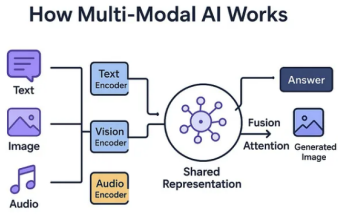


Agentic Workflows: Shifting from single prompts to autonomous agents that can use tools and execute multi-step tasks.



Data Science in 2025 — Multimodal Integration and SLMs

Native Multimodality: Models that process text, audio, image, and video simultaneously rather than using separate "wrapper" models.



Think of it as **teaching the AI multiple languages (text, vision, audio)** and then letting them talk to each other.

Real-World Examples of Multi-Modal AI

- **ChatGPT with Vision (GPT-4V)** — You can upload an image and ask questions about it.
- **Google Gemini** — Processes text, images, and audio in the same conversation.
- **CLIP (OpenAI)** — Understands how text relates to images for search and classification.
- **DALL.E 3** — Generates images from text prompts.

Small Language Models: Deploying highly capable, smaller models on-device (Edge AI) for privacy and cost-efficiency.

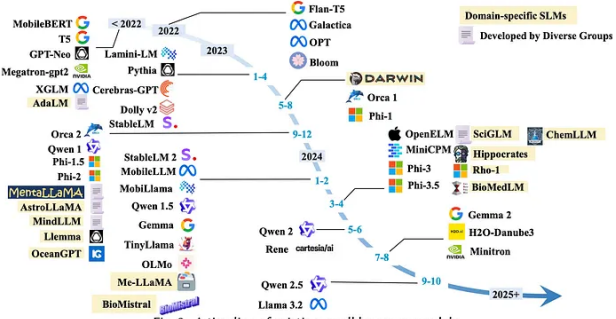


Fig. 3. A timeline of existing small language models.

AI-Native Software Engineering: Data Science merged deeper with DevOps, where AI agents began maintaining and optimizing codebases autonomously.

Data Science in 2026 — AI Bubble ?

ion Sport Culture Lifestyle The Guardian Eur

AI companies will fail. We can salvage something from the wreckage

Cory Doctorow

AI is asbestos in the walls of our tech society, stuffed there by monopolists run amok. A serious fight against it must strike at its roots

Share

I am a science-fiction writer, which means that my job is to make up futuristic parables about our current techno-social arrangements to

Economists debate if AI is fueling growth or a bubble

MSN

AI bubble issue is 'critical' to inflation view, Hildebrand says

1 hour ago

TechStock²

Davos economist warns even a 'good' AI bubble can bite in 2026

1 day ago

Law360

BoE Chief Warns Of Financial Fallout If AI Bubble Bursts

20 hours ago

Microsoft CEO Satya Nadella warns AI boom risks becoming a bubble

ortune

osoft CEO Satya ella's biggest AI bubble ing yet is a challenge t…

ours ago

DIGIT.FYI

Microsoft CEO Warns AI Risks Becoming a Speculative Bubble

43 minutes ago

ECONOMY & MARKETS

The Looming AI Bubble in 2026: Will it Pop?

By Richard Zhou

Published 01/19/26 AT 8:07 PM EST

Data Science in 2026 — Predicting is hard

Predictions made about 2025

- Elon Musk (in 2024) “AI will probably be smarter than any single human next year.”
  Evaluation: False
- Dario Amodei (Anthropic CEO, March 2025): “In 3 to 6 months . . . AI is writing 90 percent of the code.”
  Evaluation (6 months being Sep 2025): False in the relevant sense. (“Number of lines”, for example, is not a relevant metric.)

Predictions made about 2026

- Elon Musk (Jan 2025): “I think we are quite close to digital superintelligence. It may happen this year. If it doesn’t happen this year, next year for sure. A digital superintelligence defined as smarter than any human at anything.”
- Eric Schmidt: “It’s likely in my opinion that you’re gonna see world-class mathematicians emerge in the next one year that are AI based, and world-class programmers that’re gonna appear within the next one or two years.”

Hype ? Again ?

This article is more than **6 years old**

Two years until self-driving cars are on the road – is Elon Musk right?

The Tesla CEO has proclaimed that autonomous driving is a 'solved problem' but tech and executives in recent years have tempered their expectations

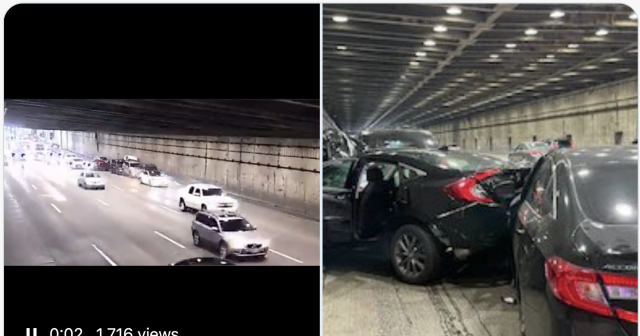
Hype ? Again ?

This article is more than **6 years old**

Two years until self-driving cars are on the road – is Elon Musk right?

The Tesla CEO has proclaimed that autono problem' but tech and executives in recen their expectations

#Tesla in 'self-driving mode' slams on brakes in tunnel for no reason. The car caused an eight-vehicle crash that injured nine people. Just hours before the crash #Musk had triumphantly announced that Tesla's "Full Self-Driving" capability was available in North America.



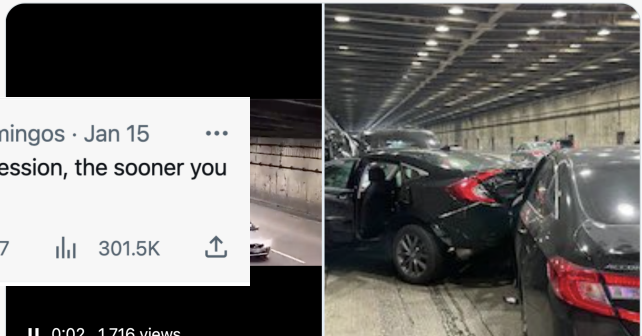
Hype ? Again ?

This article is more than **6 years old**

Two years until self-driving cars are on the road – is Elon Musk right?

The Tesla CEO has proclaimed that autono problem' but tech and executives in recen their expectations

#Tesla in 'self-driving mode' slams on brakes in tunnel for no reason. The car caused an eight-vehicle crash that injured nine people. Just hours before the crash #Musk had triumphantly announced that Tesla's "Full Self-Driving" capability was available in North America.

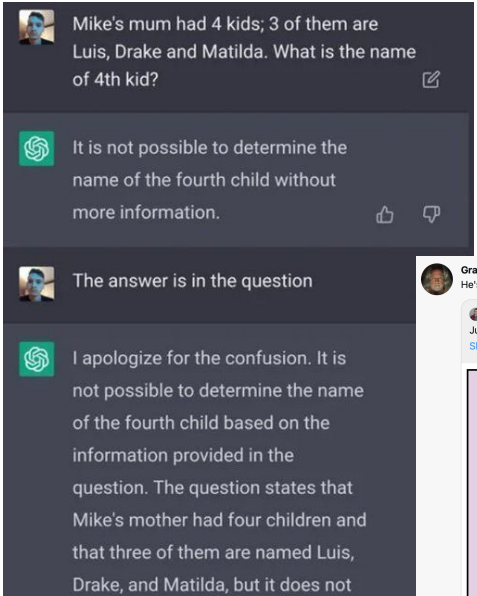


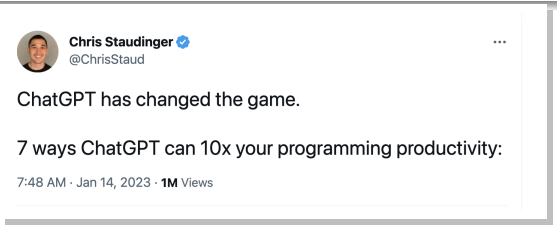
Pedro Domingos @pmddomingos · Jan 15

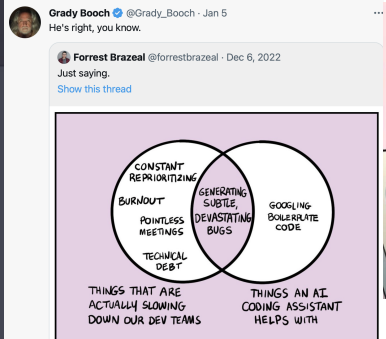
The less you know about a profession, the sooner you think it'll be replaced by AI.

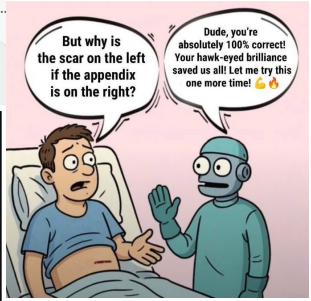
70 421 2,357 301.5K

But chatGPT/Gemini/. . . can do . . .

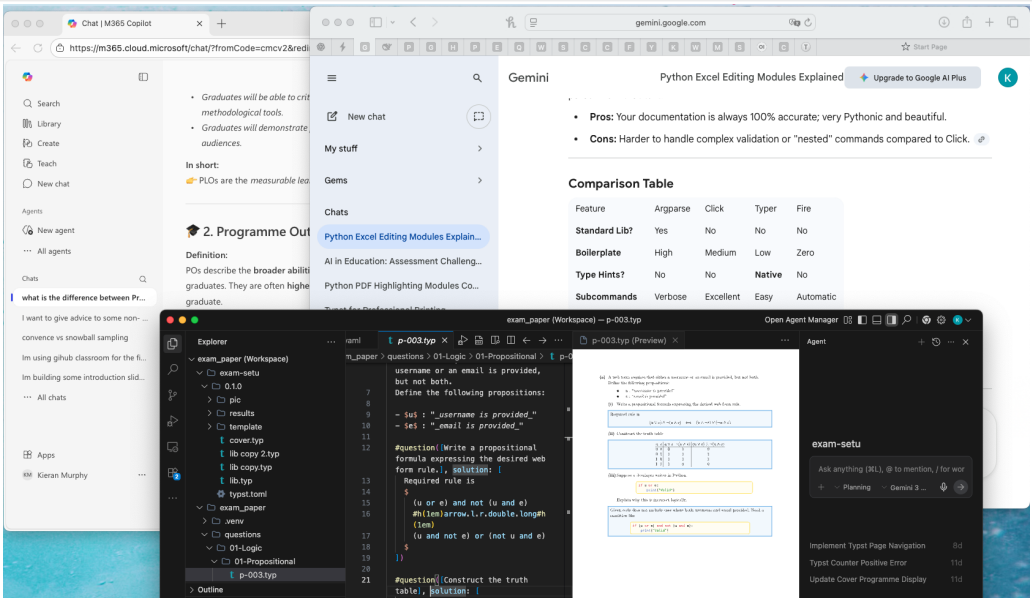








People in glasshouses should not . . .



Delivery

Resources

- All lecture slides, handouts and datasets: GitHub — setu-datamining2.github.io/live
- All activities: quizzes and assignments: Moodle — moodle.wit.ie/course/view.php?id=199957

Delivery

- Two 1-hour lectures and one 2-practical session.
  - Lecture sessions can tend to get very non-interactive so to help avoid this please ask questions.
- Slack
  - Will use this for all last minute posts and individual/group Q+A, particularly for assignments.

Strategy to handle module

- Prepare — review material in advance of the sessions, install/download the software/datasets.
- Interact — yes, this is rich coming for an introvert mathematician, but we live in strange times.
- Time management — give tasks a serious/focused effort, but when stuck ask for help.

Assessment Structure — 100% Continuous Assessment

Proposed Structure (dates open for discussion, weights ±5%)

- Applied AI Assignment — Build a Studyclix application for past Discrete Mathematics exam papers. (pandas and friends)
  - Given Week 1, due in Week 6, grading and validation Week 7&8.
  - 35%
- Kaggle Assignment
  - Given Week 4, due in Week 14, grading and validation Week 15.
  - 45%
- Practical and Quizzes
  - Nothing sorted yet — guess 2 practicals and 2 quizzes sometime during the semester.
  - 20%

Calendar

- Week 14/15 end of semester individual review interview (similar to S1).
- $\underbrace{\text{4 weeks} + \text{reading week} + \text{5 weeks} + \text{Easter break (2 weeks)} + \text{3 weeks}}_{\text{12 teaching weeks}} + \text{3 weeks for CA}$

Assessment — Grading vs Validation

Grading

Does the assignment meet the stated specifications?

- Relatively easy to check against a rubric

Validation

Is this the student's work?

- Traditionally, could use criteria "Did you write this?". (No longer meaningful with generative AI.)
- Use "understanding" as proxy for originality.
  - $\Longrightarrow$ I need to do interviews/Q&A after submission.
  - $\Longrightarrow$ You need to ensure that you understand what you submit and what effect each line has.
- Is work across in-class activities comparable to assignments?

Outline

Three Components of a Machine Learning Problem

It is easy to get lost among the multitude of choices one needs to make when given data mining problem. A good decomposition is the following:

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

[†] A Few Useful Things to Know about Machine Learning, Domingos, 2012.

3 Components — Representation

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search

Representation refers to formulating the problem as a machine learning problem — typically a classification problem, a regression problem or a clustering problem.

- How do we represent the input?
- What features to use?
- How do we learn additional features?
- With each type of problem, we have multiple subtypes.
  For example which classifier? a decision tree, a neural network, a support vector machine, a hyperplane that separates the two classes etc.

3 Components — Evaluation

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search

Evaluation refers to an objective function or a scoring function, to distinguish good models from a bad model.

- For a classification problem, we need this function to know if a given classifier is good or bad. A typical function can be based on the number of errors made by the classifier on a test set, using precision and recall.
- For a regression problem, it could be the squared error, or likelihood. Do we include regularisation? etc

3 Components — Optimisation

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search

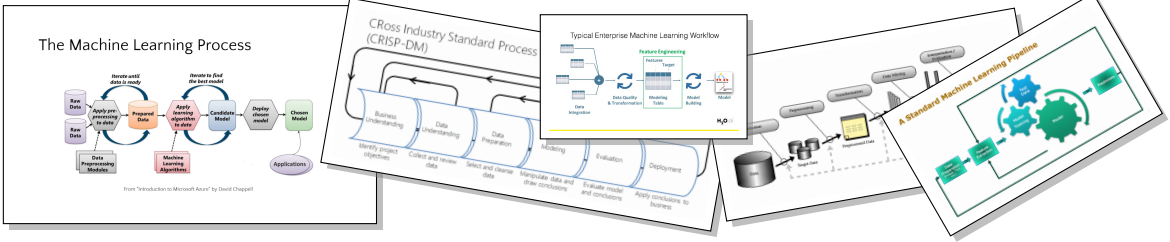
Optimisation is concerned with searching among the models in the language for the highest scoring model.

- How do we search among all the alternatives?
- Can we use some greedy approaches, branch and bound approaches, gradient descent, linear programming or quadratic programming methods.

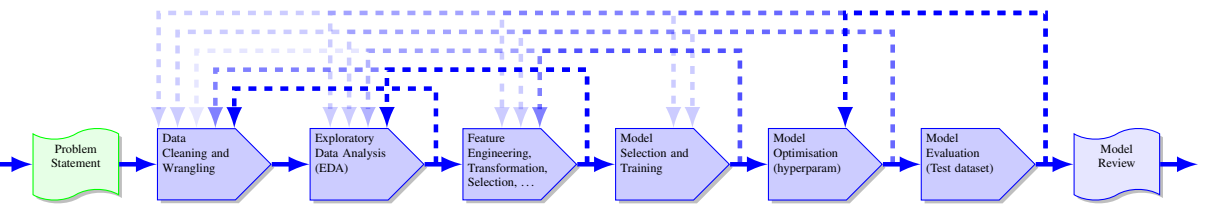
Outline

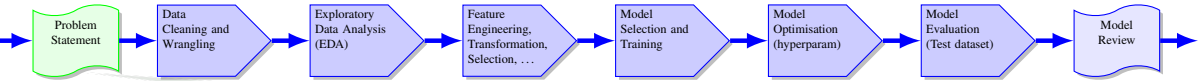
Data Mining Workflow

There are many, many . . .



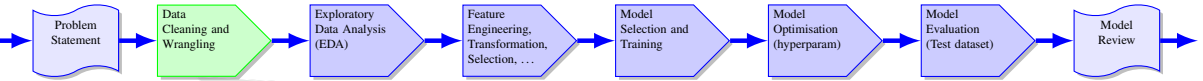
So why not make YADMW (Yet Another Data Mining Workflow)?





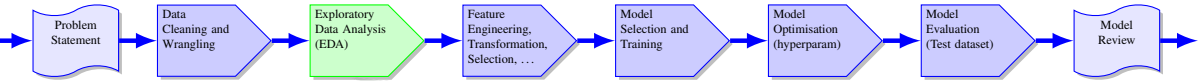
What type of problem is it?

- Exploratory data analysis
  Do we just want to see what the data says?
- Association / Rule finding
  Are we searching for relations/patterns?
- Hypothesis testing (Statistical)
  Do we have a theory we wish to test?
- Model building
  Do we wish to build a representation of some pattern within the data?
  - Supervised ⇐ data spilt into input variables (features) and output variable(s) (target(s))
    - Classification (target is categorical) vs regression (target is continuous)
  - Unsupervised ⇐ no target
    - Clustering — grouping similar cases

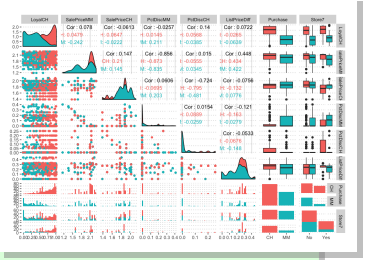


How to import and prepare data for subsequent analysis/processing?

- Multiple file formats
  - Pandas supports a wide collection of file formats but default options often need to be changed to suit data.
  - Main file format (Comma Separated Values (csv)) does not support meta-data, is slow, and results in large files $\Longrightarrow$ use other formats (pickle, feather) to store datasets between steps in the workflow.
- Assumptions made by input parser can be important (i.e., bite you when you least expect)
  - Scientists rename human genes to stop Microsoft Excel from misreading them as dates
  - Pandas vs excel use different heuristics to decide on data type of each variable.
- Sub-tasks
  - Check dimension (number of rows/cases, number of columns/variables).
  - Check data types (categorical, ordinal, or numerical (discrete/continous)) of each variable.
  - Check for missing values, encoding errors, etc.
  - Merge tables, apply filters, and general data wrangling to generate (tabular) dataset suitable for EDA.

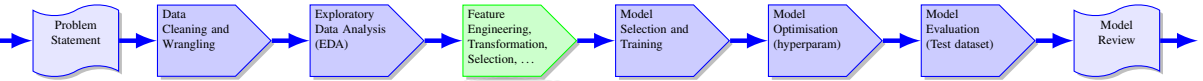


What is the data telling us?



- Univariate descriptive statistics — examine each variable
  - What are typical values?
  - What is the variation / spread / range?
  - What does the data look like …bell curve, bath tub curve, etc. ?
- Bi- / multi- variable descriptive statistics
  - Identifying relationships between variables.
- All descriptive statistics methods summarise data:

✔ A summary is good since it helps to focus on simpler and important aspects.
✘ A summary is bad if it focuses on irrelevant or the wrong aspects.
⇒ Need to use multiple methods, be aware of their strengths/deficiencies.

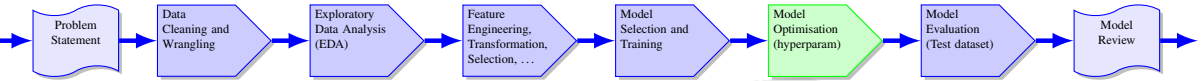


Can we transform, encode/bin, select, ..., the given features to improve model training?

- Better features can mean:
  - Better model performance and reduce training times.
  - Simpler models become applicable — think linear/logistic regression.
  - More explainable models — the future of machine learning (hopefully).
  - Cheaper and easier models to deploy.
- Feature selection reduces the number of features used in the model:
  - Drop features that have low variability.
  - Drop features that have no relation to target.
  - Drop features that are highly related to other features — multicollinearity.
  - Keep features whose addition to model have the largest improvement in model score.
- Feature extraction merges existing features to generate (hopefully) fewer features with essentially all the variation.

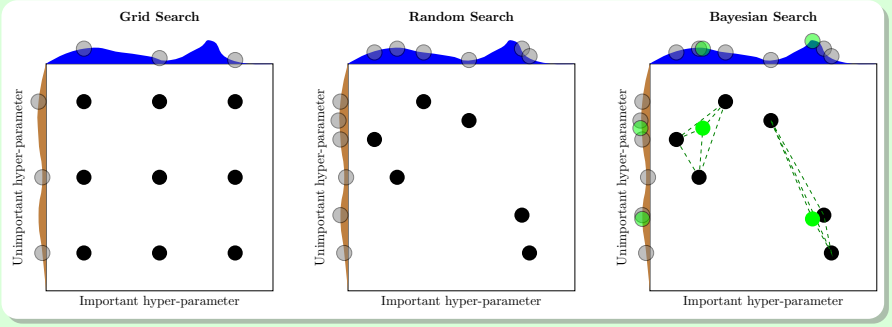
Which models are suitable?

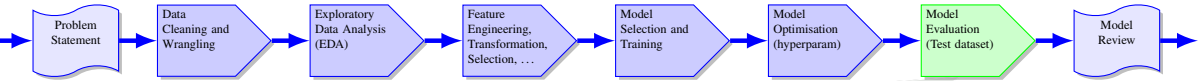
- Models vary greatly in terms of capabilities/deficiencies — usually aim to build a short list of candidate models, which are subsequently optimised in the next step.
- Select models based on different algorithms/approaches.
- Select (loss function and) evaluation metric.
  - Loss function is used to train model, evaluation metric is used to evaluate model (post training).
- Relative model performance can help identify issues with data.
  - Outliers can negatively affect linear regression but have smaller impact on decision tree based models.



How do we determine optimal values of the hyper-parameters?

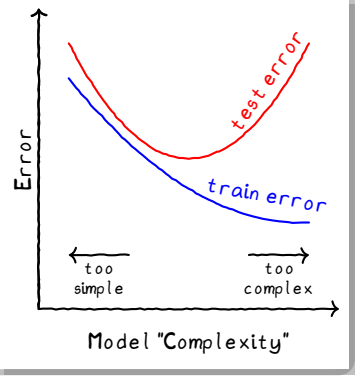
- Most models have options which control how a model "learns" from the training data.
- Three search stratagies: Grid search $<$ Random search $\ll$ Bayesian search

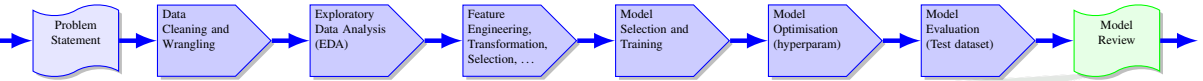




How well does the model generalise (to unseen data)?

- In the machine learning approach (vs statistical approach) we rely on model performance on **unseen data** to evaluate models.
  - Split data into train/test, only use train dataset for all modelling decisions.
  - Data leakage (MachineLearningMastery article), where information outside the train dataset is used in model building.
- Is there evidence for overfitting?
  - Does the model perform much better on training dataset than on the test dataset?
- Multiple techniques to address overfitting:
  - Regularisation (linear / logistic regression).
  - Trimming (decision trees).
  - Dropout (neural networks), Batch normalisation (CNN).





How well have we addressed the problem statement?

- A what level of accuracy (or other metrics) does a model become useful?
  - This is a business, medical, … decision
  - The larger the relative payoff the weaker the model can be and still be useful.
- OK, finally ready to implement/deploy model …
  - Separate skillset / concerns
  - MLOps = ML + DevOps
  - Monitoring of model drift needed.
- towards data science What is MLOps — Everything You Must Know to Get Started

Q: Why don't we automate all of this shtuff?
Tools are getting better and easier to use, but need intervention/direction (data can be weird in weird ways)

– xkcd.com/2054