MATH608/MATH457 Practical and Innovative Analytics in Data Science

Szu-Chi Chung

Department of Applied Mathematics, National Sun Yat-sen University

Lectures

- Class hours: Mon. (9:10-12:00)
  - Classroom: 理 SC-2004
- Lecture: Szu-Chi Chung (鍾思齊)
  - Office: 理 SC 2002-4
  - Office hours: Mon. 16:00~18:00 and Wed. 16:00~18:00
- T.A.: 李祐瑄、孫瑞鴻
  - Office: 理 SC 2003-2
  - TA hour: Thur. 15:00~17:00
- Course website: https://phonchi.github.io/nsysu-math608/
- Facebook: https://www.facebook.com/groups/464200246458564

Textbook and requirement

- Textbook: *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition*
  - Authors: Aurélien Géron
  - https://github.com/ageron/handson-ml3
- Deep Learning for Coders with Fastai and PyTorch: AI Applications Without a PhD
  - Authors: Jeremy Howard and Sylvain Gugger
  - https://github.com/fastai/fastbook2e
- For the data processing reference book: *Python Data Science Handbook*
  - Authors: Jake VanderPlas
  - https://github.com/jakevdp/PythonDataScienceHandbook

Textbook and requirement

- You should have basic knowledge about statistics and modeling
  - Please refer to our course website for resources and MOOC to review the basic concepts if needed
    - https://phonchi.github.io/nsysu-math608/materials/
    - https://phonchi.github.io/nsysu-math524/

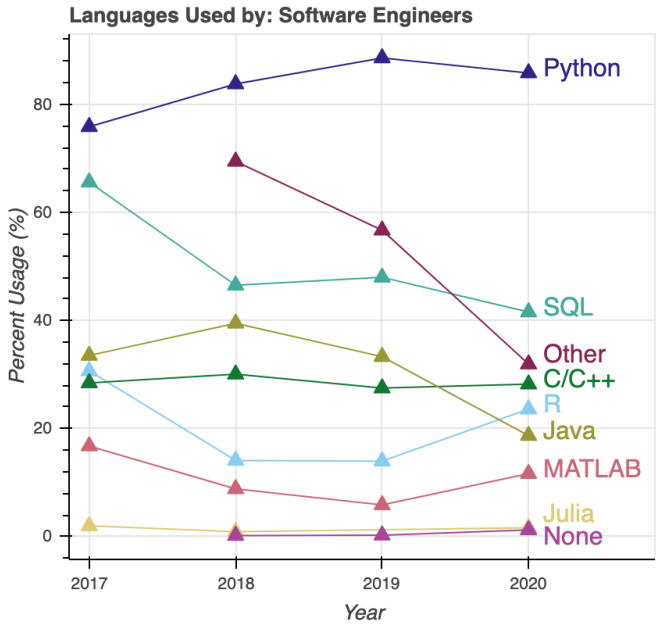
- Programming language: Python
  - You are asked to use python to implement the assignment, midterm and final
  - Since it is the most popular language in the field of data science
  - It is free and easy to learn
  - The homework and related material will be available in the course website

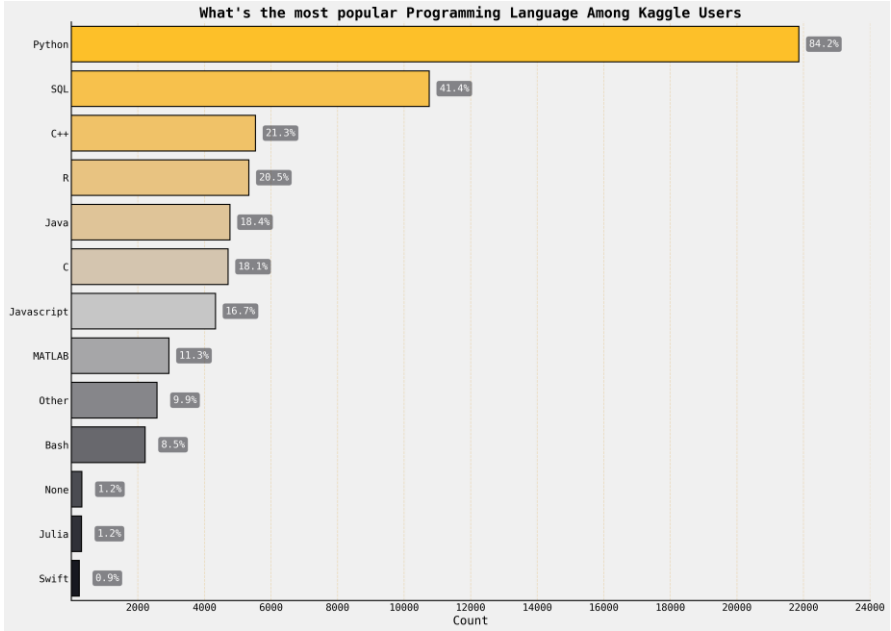
Grading policy

- Grading
  - Homework 20% (Both conceptual and coding part, about 4~5 times)
  - Midterm project 40% (You are free to choose any dataset and any analysis method)
  - Final project 40% (We will provide a dataset and it will be a data-centric competition)
- Midterm project:
  - Organize a team of 2~3 persons
  - Presentation will be held on 10/28
  - Must hand in a report
  - Score will be the summation of students (15%), TA(15%) and lecturer (10%)
- Final project:
  - Organize a team of 2~3 persons
  - Must hand in a report

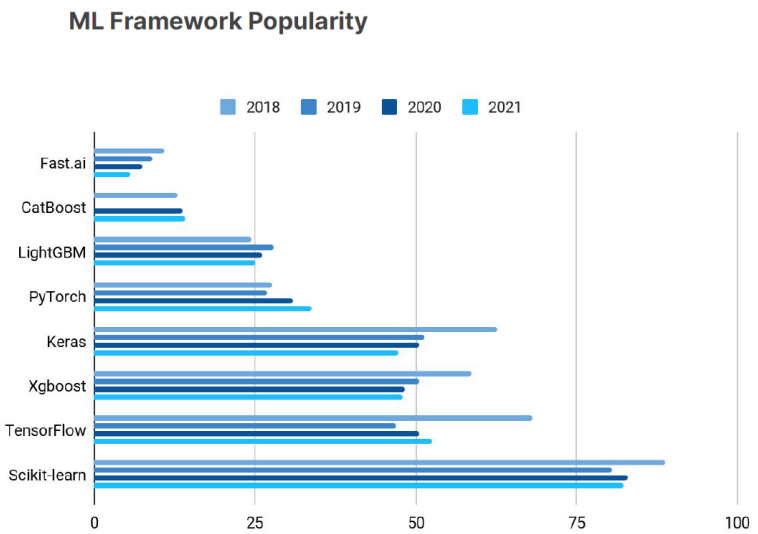
Dataset and competition

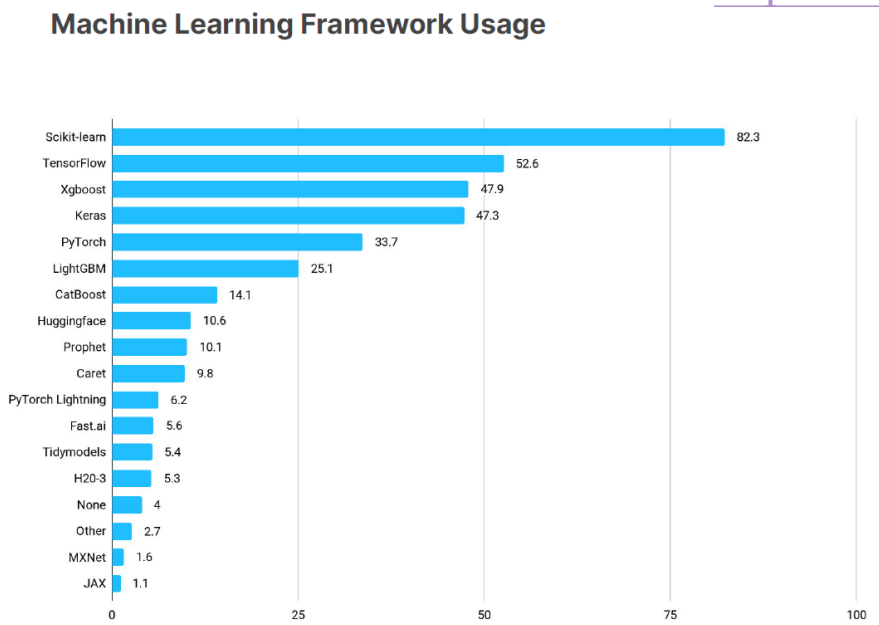
- Dataset
  - Dataset search platform provided by National Development Council
    - https://data.gov.tw/
  - Kaggle
    - https://www.kaggle.com/datasets
  - Google dataset search
    - https://datasetsearch.research.google.com/
  - Paperwithcode
    - https://paperswithcode.com/datasets
- Competition
  - Kaggle
    - https://www.kaggle.com/competitions
  - Tbrain
    - https://tbrain.trendmicro.com.tw/





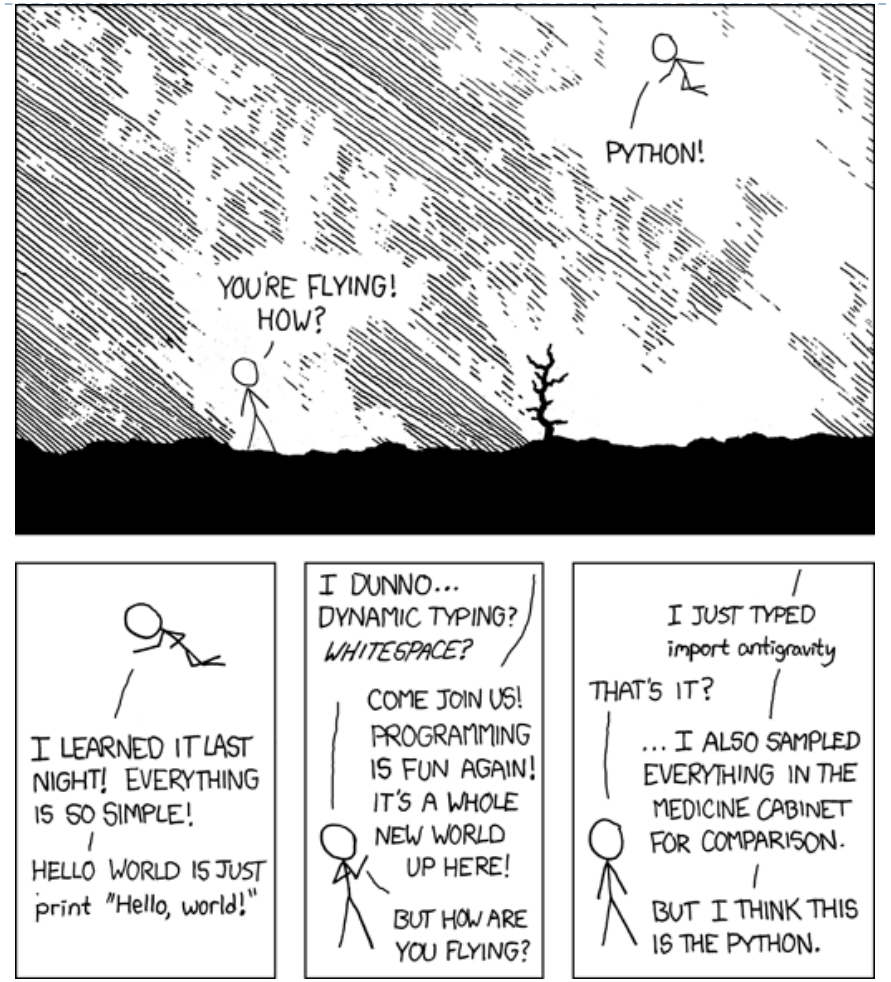
https://www.kaggle.com/kaggle-survey-2021





Learning Python

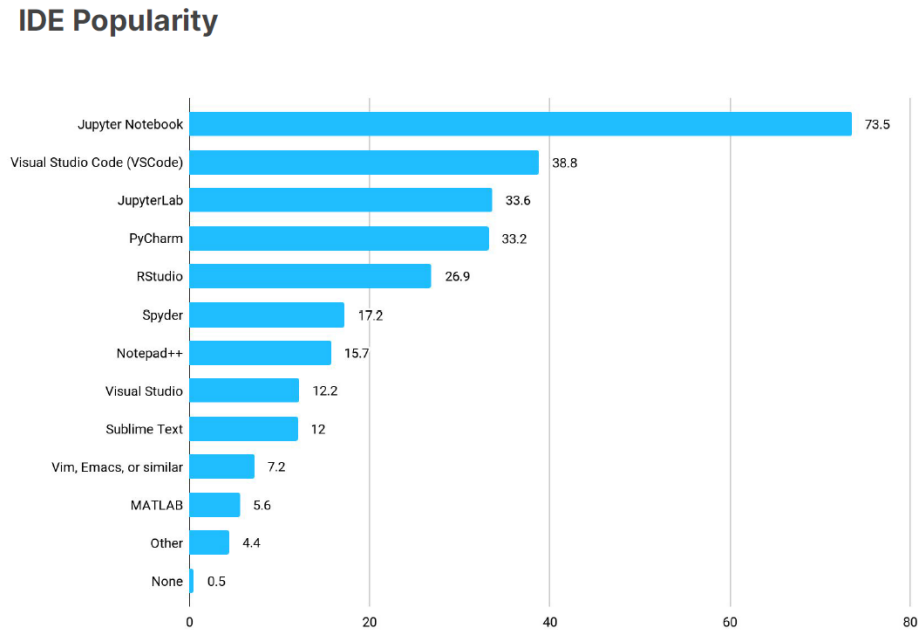
- Python
  - **Learn X in Y minutes**
  - Kaggle Python tutorial
  - Python for Everybody
- Python scientific computing
  - https://sites.google.com/view/nlpcolearningnsysu/support?authuser=0#h.cau5zoy87l0z
  - https://scipy-lectures.org/
  - More
- Learning with AI
  - Instructions
  - ChatGPT, Gemini or Copilot

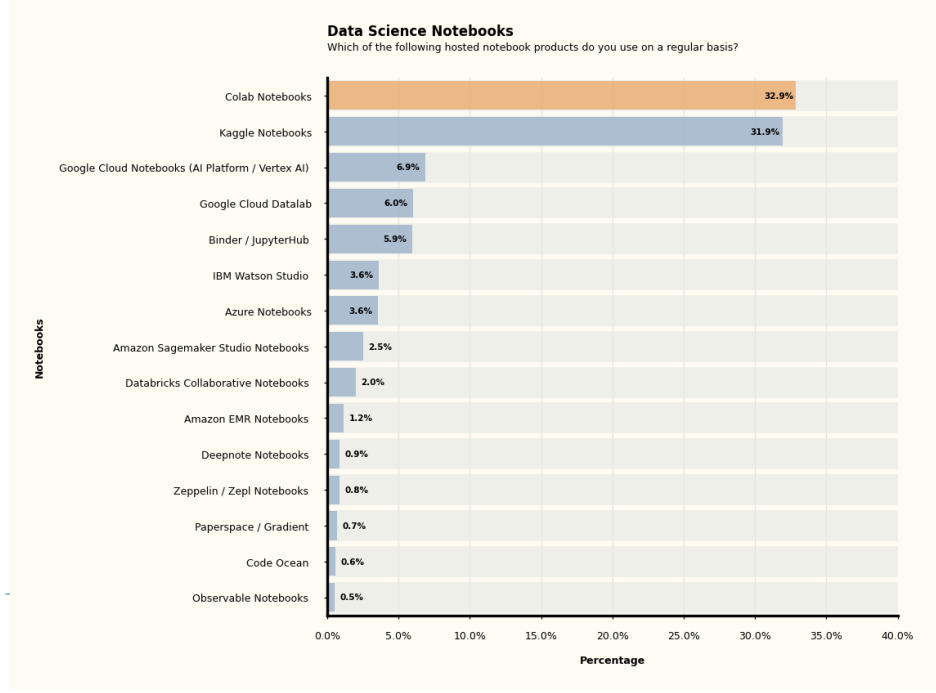


https://xkcd.com/353/

Environment

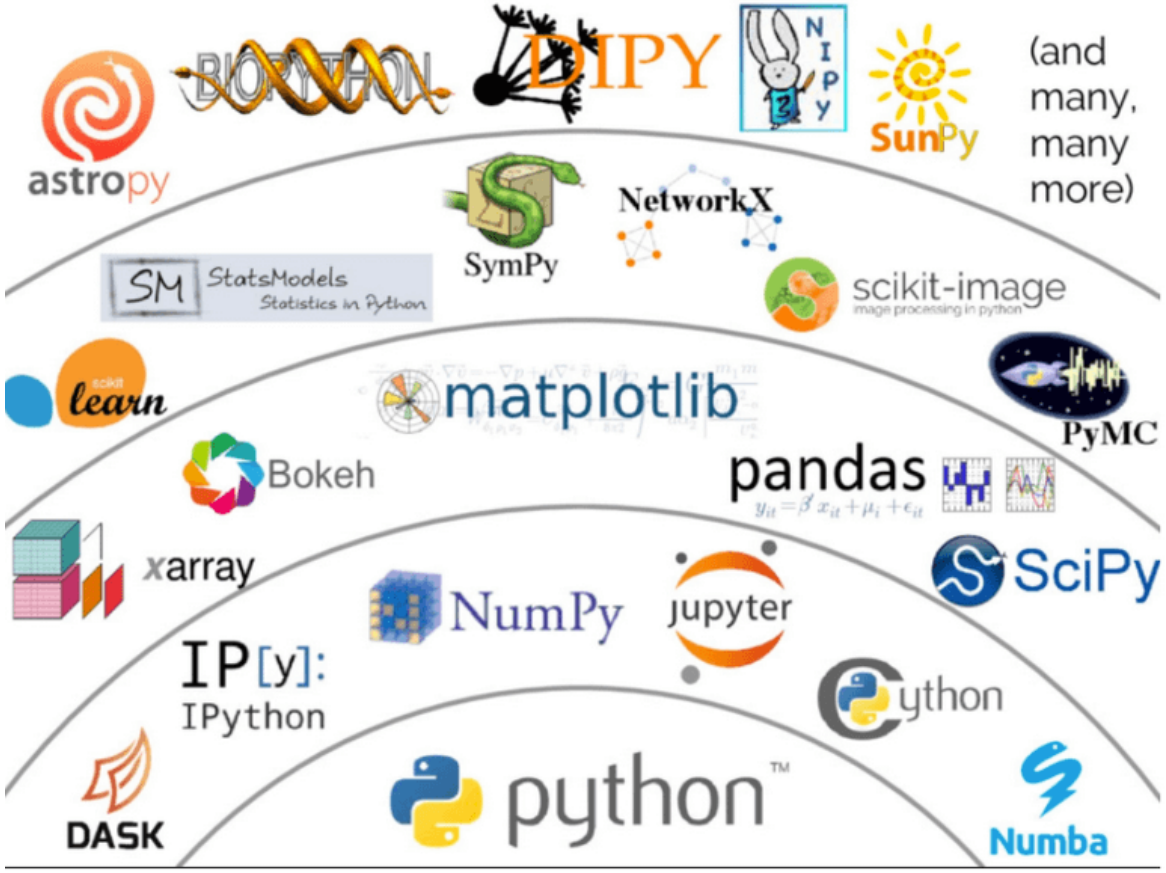
- Jupyter notebook
  - Colab - https://colab.research.google.com/
  - Kaggle - https://www.kaggle.com/docs/notebooks
  - Jupyterlab - https://www.anaconda.com/products/individual
- Markdown (Use on https://hackmd.io/, GitHub, Jupyter notebook...)
  - Learning
    - https://commonmark.org/
    - https://learnxinyminutes.com/docs/markdown/
- Cloud service
  - Google computing platform





The Pydata Stack

- In 2017, a keynote at PyCon presented a schematic of the scientific Python stack
  - Project Jupyter and IPython for interactive computing and IDEs
  - NumPy for numerical array computing
    - Numba for just-in-time compilation
    - Cython for ahead-of-time compilation
  - Pandas for dataframe (Labeled array)
  - Scikit-learn and Statsmodel for modeling
  - Seaborn for visualization
- Install Anaconda
  - https://www.anaconda.com/products/individual
- Checkout https://rapids.ai/ for gpu-accelerated computing



Source: https://coiled.io/pydata-dask/

Our aim

- This course focuses on the practical aspect of data science in the real world. In the course, students will learn to engage in a real-world project requiring them to apply skills from the entire data science pipeline: preparing, organizing, and transforming data, constructing a model, and evaluating results.
- Moreover, high-level descriptions of how to apply deep learning for computer vision and natural language problems will also be covered

Related to other course

- More theoretical foundation
  - Mathematical statistics, statistical inference or principles of artificial intelligence
- More about modeling
  - Statistical learning and data mining, machine learning or deep learning, machine learning with python
- Apply to a specific domain and advance modeling
  - Time series analysis or survival analysis
- Implement from scratch
  - Python and machine learning algorithms or https://dafriedman97.github.io/mlbook/content/introduction.html
- High-performance (Parallel) computing, Database management and systems…

Schedule

週次 Week	日期 Date	授課內容及主題 Content and topic
1	2024/09/08~2024/09/14	The data science landscape
2	2024/09/15~2024/09/21	Framing the problem and constructing the dataset
3	2024/09/22~2024/09/28	Data wrangling and relational database
4	2024/09/29~2024/10/05	Data cleaning and feature engineering
5	2024/10/06~2024/10/12	Feature selection and extraction
6	2024/10/13~2024/10/19	Explainable AI
7	2024/10/20~2024/10/26	Model serving
8	2024/10/27~2024/11/02	Midterm project
9	2024/11/03~2024/11/09	The deep learning journey
10	2024/11/10~2024/11/16	Image processing with Convolutional Neural Networks
11	2024/11/17~2024/11/23	Sequence processing using Recurrent Neural Network
12	2024/11/24~2024/11/30	Transfer learning and self-supervised learning
13	2024/12/01~2024/12/07	Representation learning
14	2024/12/08~2024/12/14	Hyperparamter search and experiment management
15	2024/12/15~2024/12/21	Final project
16	2024/12/22~2024/12/28	Final project
17	2024/12/29~2025/01/04	Flexible learning
18	2025/01/05~2025/01/11	Flexible learning

End-to-End Data Science Project

Szu-Chi Chung

Department of Applied Mathematics, National Sun Yat-sen University

Data

- We live in a world that's drowning in data
  - Websites track every user's every click
  - Your smartphone is building up a record of your location every second of every day
  - People wear smart watch that are always recording their heart rates, movement habits, diet, and sleep patterns
  - Smart cars collect driving habits, smart homes collect living habits, and smart marketers collect purchasing habits
- The internet itself represents a huge graph of knowledge that contains an enormous cross-referenced encyclopedia; domain-specific databases about movies, music, sports results…

Data

- Facebook asks you to list your hometown and your current location, ostensibly to make it easier for your friends to find and connect with you. But it also analyzes these locations to identify global migration patterns and where the people in different clubs live!

- As a large retailer, Costco tracks your purchases and interactions, both online and in-store. And it may, for example, uses the data to predict which of its customers are pregnant, to better market baby-related purchases to them!

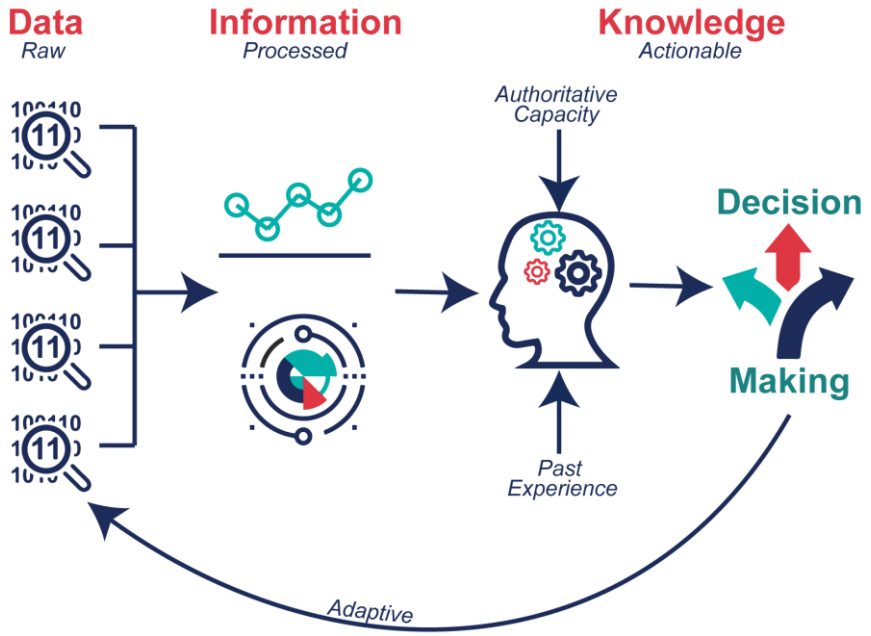
- Some others also use data to make government more effective, to help the homeless, and to improve public health

https://www.facebook.com/photo/?fbid=532707412287717&set=a.422136660011460

What Is Data Science?

- Data science is an interdisciplinary field that uses scientific methods, algorithms and systems to extract knowledge and insights from noisy, structured and unstructured data, and apply knowledge and actionable insights from data

- A data scientist often creates programming code, and combines it with statistical knowledge to create insights from data



https://oddur-bjarnason.org/the-data-information-knowledge-process/

What Is Data Science?

- The field encompasses formulating data science problems, preparing data for analysis, analyzing data, developing data-driven solutions, and presenting findings to inform high-level decisions
- As such, it incorporates skills from computer science, statistics, information science, mathematics, information visualization, graphic design, complex systems, communication and business
- In 2015, the American Statistical Association identified database management, statistics and machine learning, and distributed and parallel systems as the three emerging foundational professional communities of data science
  - Put attention on data preparation - https://leemeng.tw/why-you-need-to-learn-data-engineering-as-a-data-scientist.html

Who Are Data Scientists?

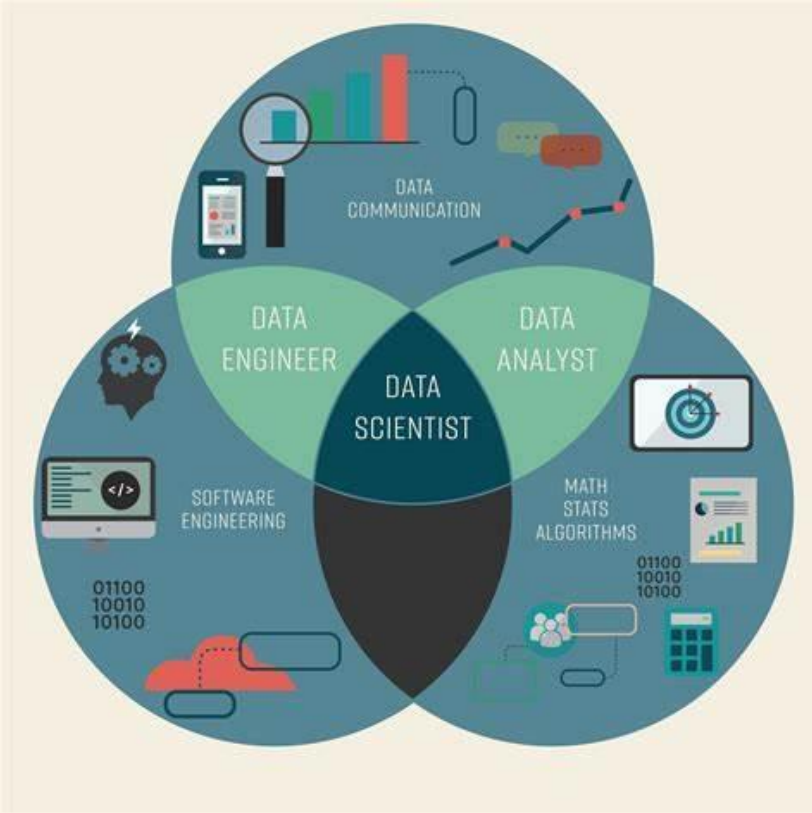
- There's a joke that says a data scientist is someone who knows more statistics than a computer scientist or more computer science than a statistician

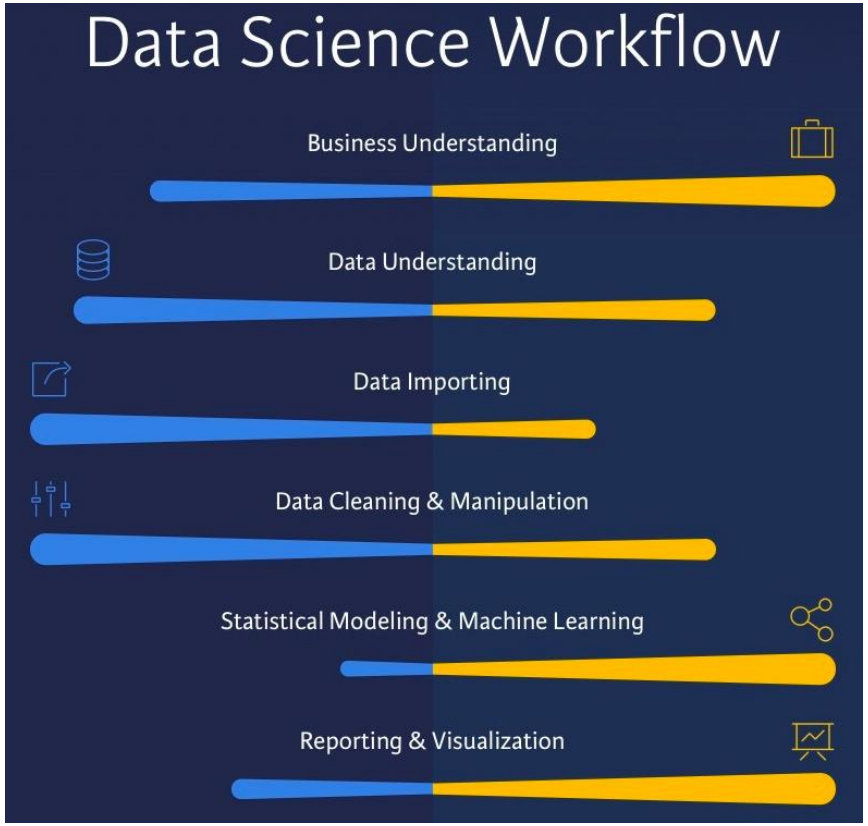
- In fact, some data scientists are statisticians, some are software engineers, some are …

- We'll say that a data scientist is someone who extracts insights from messy data. Today's world is full of people trying to turn data into knowledge!

Who Are Data Scientists?

- Data scientist

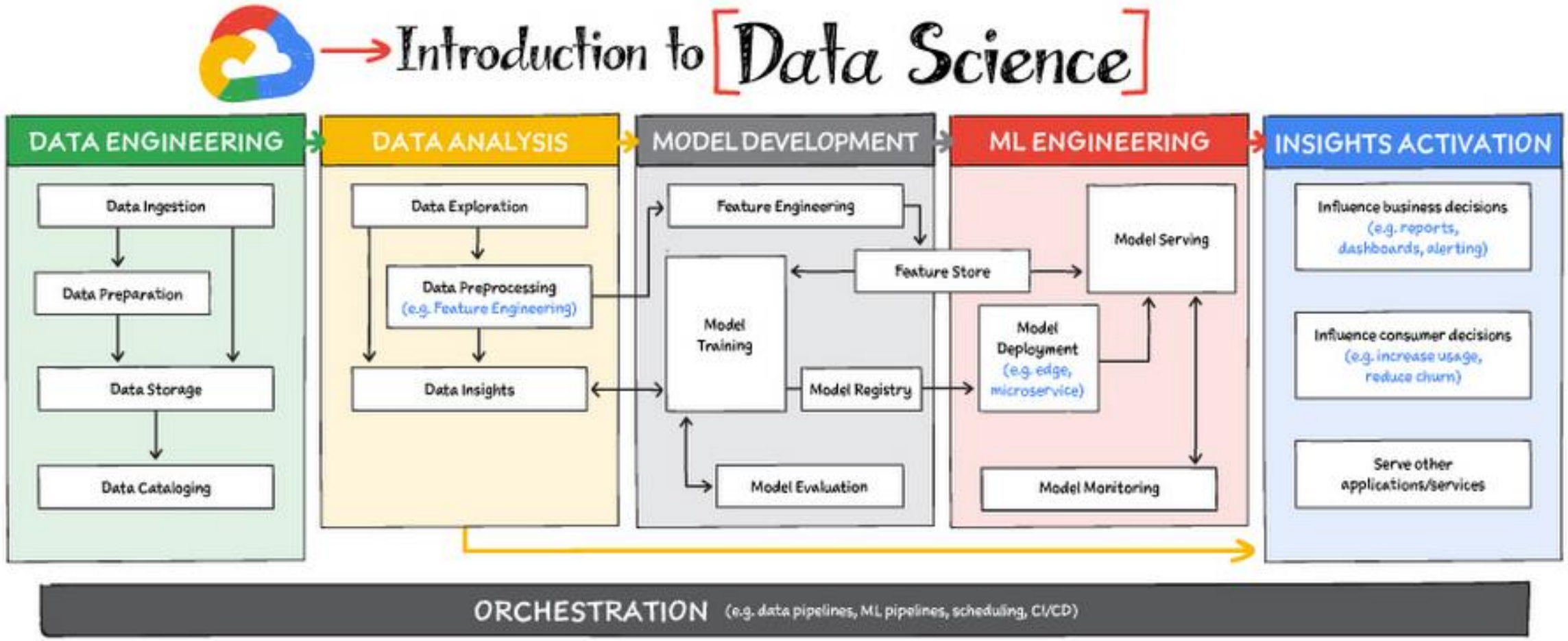




https://www.datacamp.com/community/blog/data-scientist-vs-data-engineer
https://k21academy.com/microsoft-azure/data-science-vs-data-analytics-vs-data-engineer/

- Data scientists roadmap
  - https://github.com/AMAI-GmbH/AI-Expert-Roadmap

The Pipeline



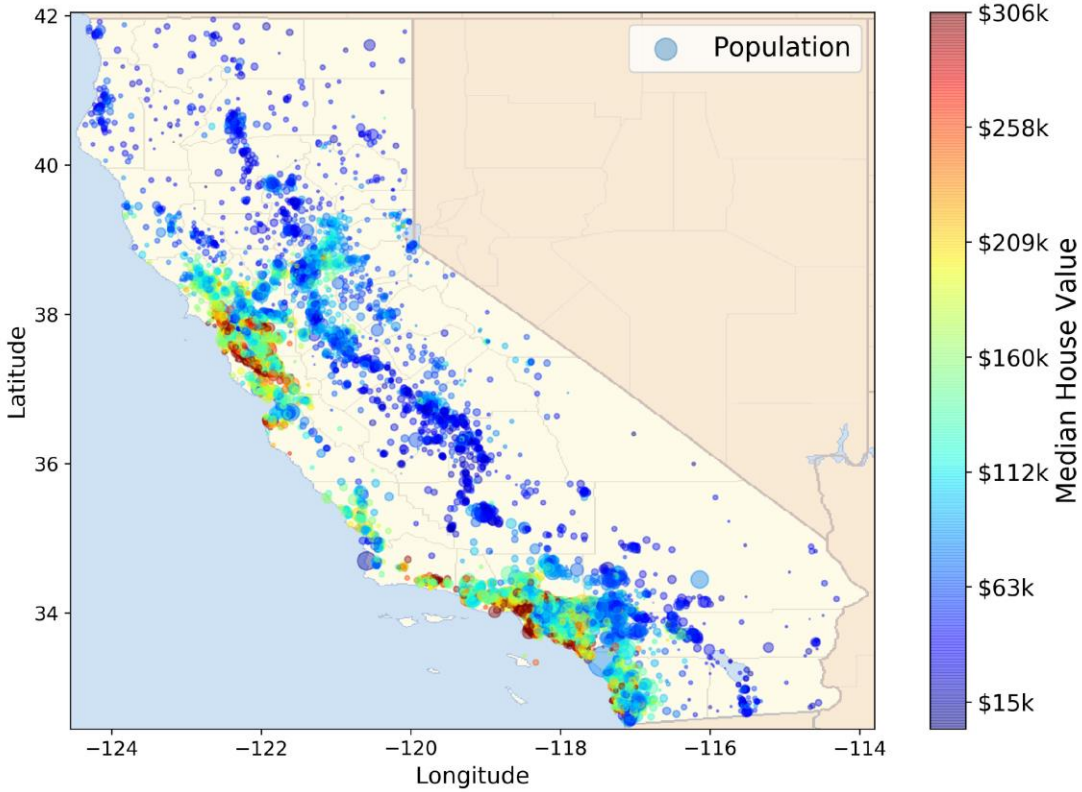
https://cloud.google.com/blog/topics/developers-practitioners/intro-data-science-google-cloud

The Checklist

1. Look at the big picture
2. Get the data
3. Discover and visualize the data to gain insights
4. Prepare the data for Machine Learning algorithms
5. Select a model and train it
6. Fine-tune your model
7. Present your solution and explain your model
8. Launch, monitor, and maintain your system

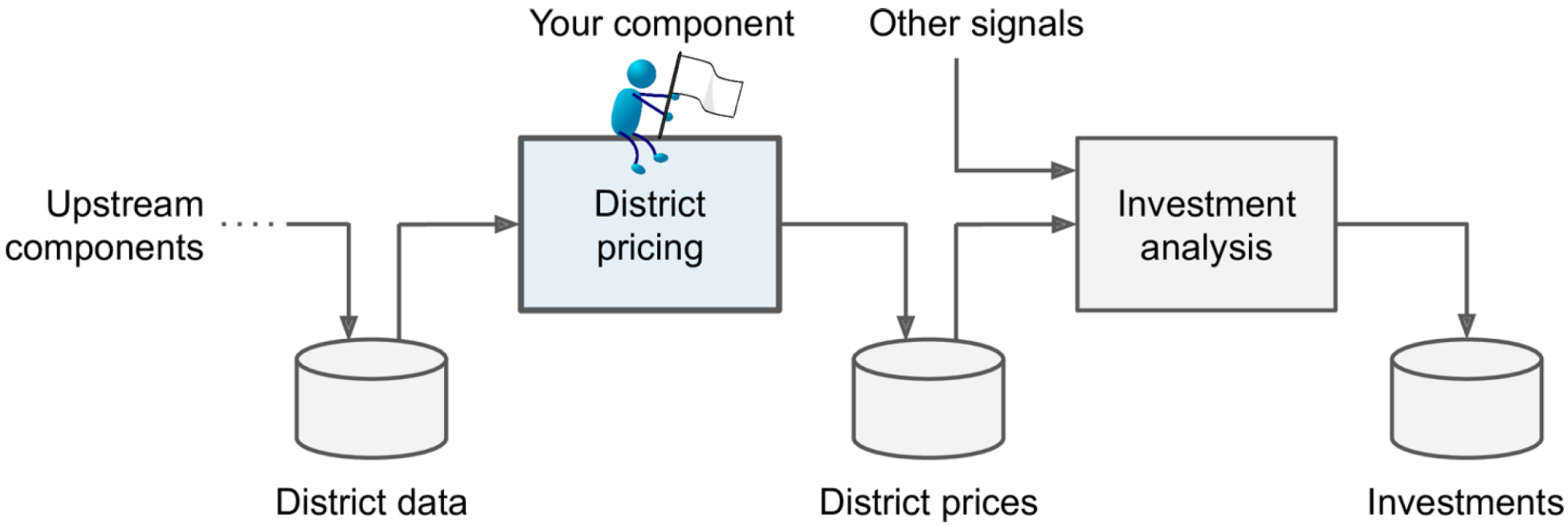
1. Look at the Big Picture

- Your first task is to use California census data to build a model of housing prices in the state
  - This data includes metrics such as the population, median income, and median housing price for each districts in California
  - Your model should learn from this data and be able to predict the median housing price in any district, given all the other features



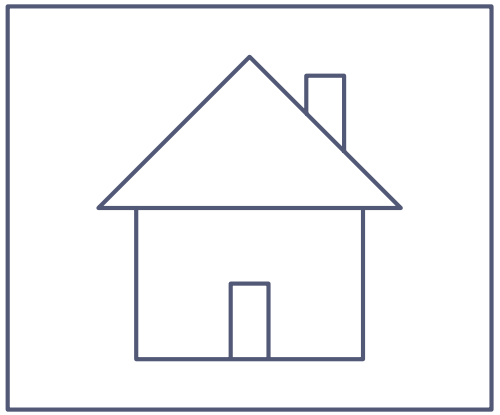
1. Look at the Big Picture – Business objective

- How does the company expect to use and benefit from this model?
  - Your model's output (a prediction of a district's median housing price) will be fed to another ML system, along with many other signals. This downstream system will determine whether it is worth investing in a given area or not



1. Look at the Big Picture

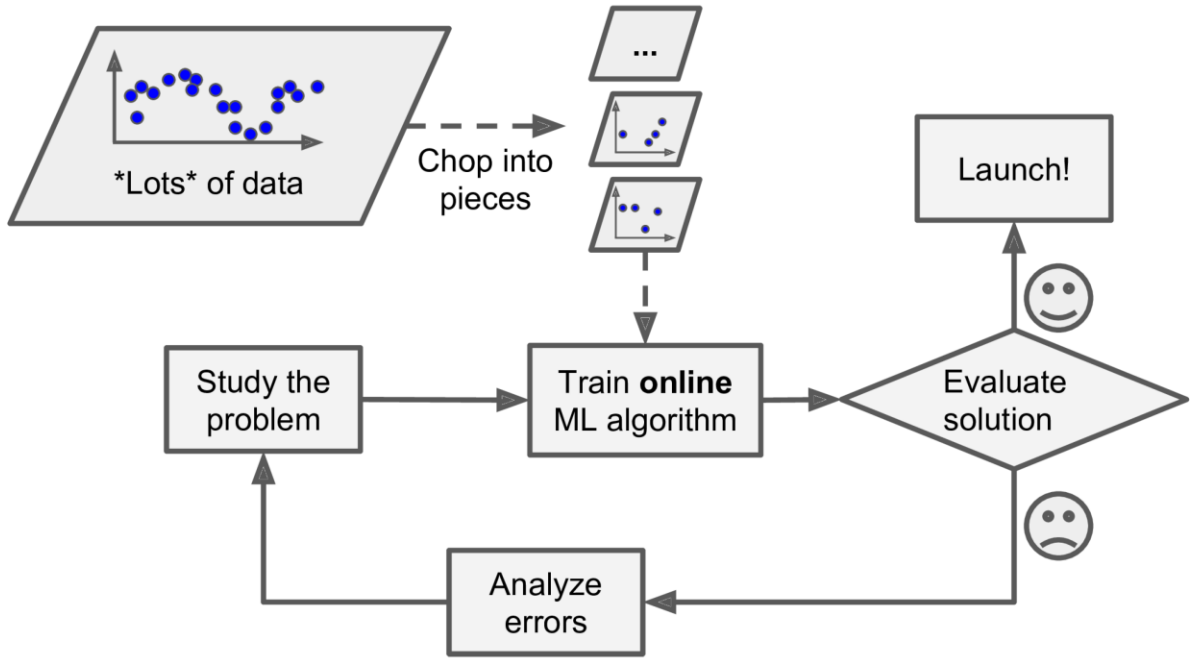
- What the current solution looks like (if any)?
  - The district housing prices are currently estimated manually by experts: a team gathers up-to-date information about a district, and when they cannot get the median housing price, they estimate it using *complex rules*
  - This is costly and time-consuming, and their estimates are not great which often off by more than 30%! This is why the company thinks that it would be useful to train a model to predict a district's median housing price, given other data about that district



1. Look at the Big Picture - Frame the Problem

- What kind of task?

1. It is a typical *supervised learning* task, since you are given labeled training examples
2. It is also a *multiple regression* task and a *univariate regression* problem, since we are using multiple features and trying to predict single value for each district
3. There is no particular need to adjust to changing data rapidly, and the data is small enough to fit in memory, so plain *batch learning* should do just fine



1. Look at the Big Picture - Select a Performance Measure

- A typical performance measure for regression problems is the Root Mean Square Error (RMSE) or l_2 norm

$$RMSE(X,h) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(h(x^{(i)}) - y^{(i)})^2}$$

  - Suppose that there are many outlier districts. In that case, you may consider using the mean absolute error

$$MAE(X,h) = \frac{1}{n}\sum_{i=1}^{n}|h(x^{(i)}) - y^{(i)}|$$

  - The higher the norm index, the more it focuses on large values and neglects small ones. This is why the RMSE is more sensitive to outliers than the MAE

- What would the minimum performance needed?

1. Look at the Big Picture - Check the Assumptions

- Lastly, it is good practice to list and verify the assumptions that have been made so far (by you or others)
  - This can help you catch serious issues early on. For example, the district prices that your system outputs are going to be fed into a downstream ML system, and you assume that these prices are going to be used as such
  - But if the downstream system converts the prices into categories (e.g., “cheap,” “medium,” or “expensive”) and then uses those categories instead of the prices themselves? If that’s so, then the problem should have been framed as a classification task, not a regression task!

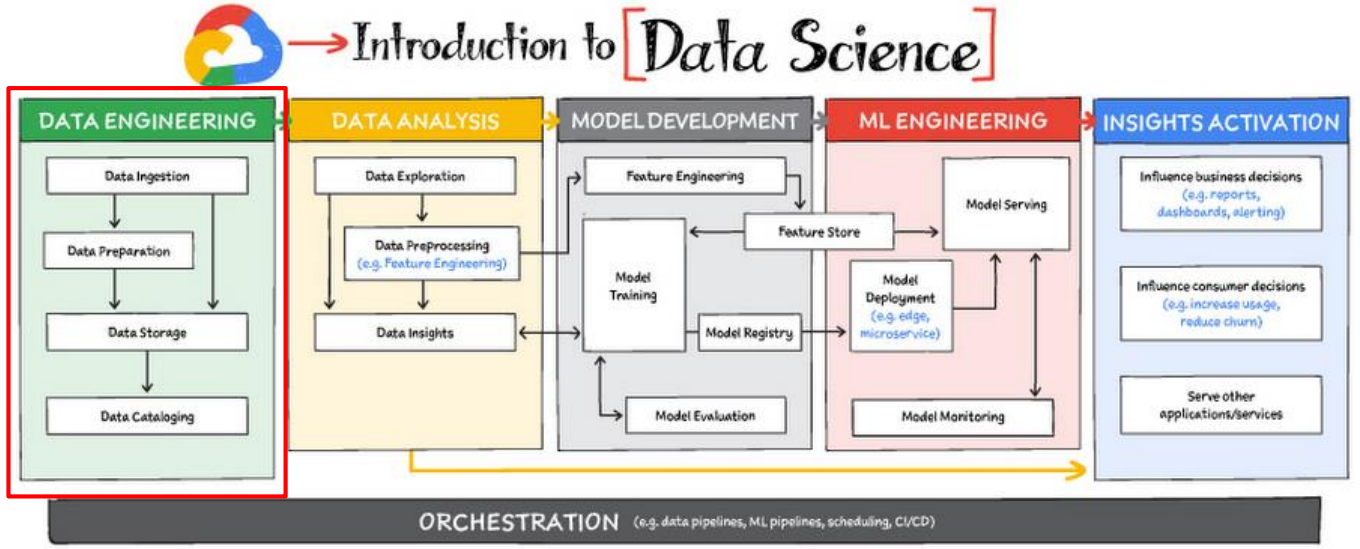
2. Get the Data

- Find and get the data
  - List the data you need and how much you need
  - Find and document where you can get the data
  - Check how much space it will take

- In typical environments your data would be available in a relational database (or some other common data store) and spread across multiple tables/documents/files
  - To access it, you would first need to get your credentials and access authorizations and familiarize yourself with the data schema
  - You should check legal constraint here

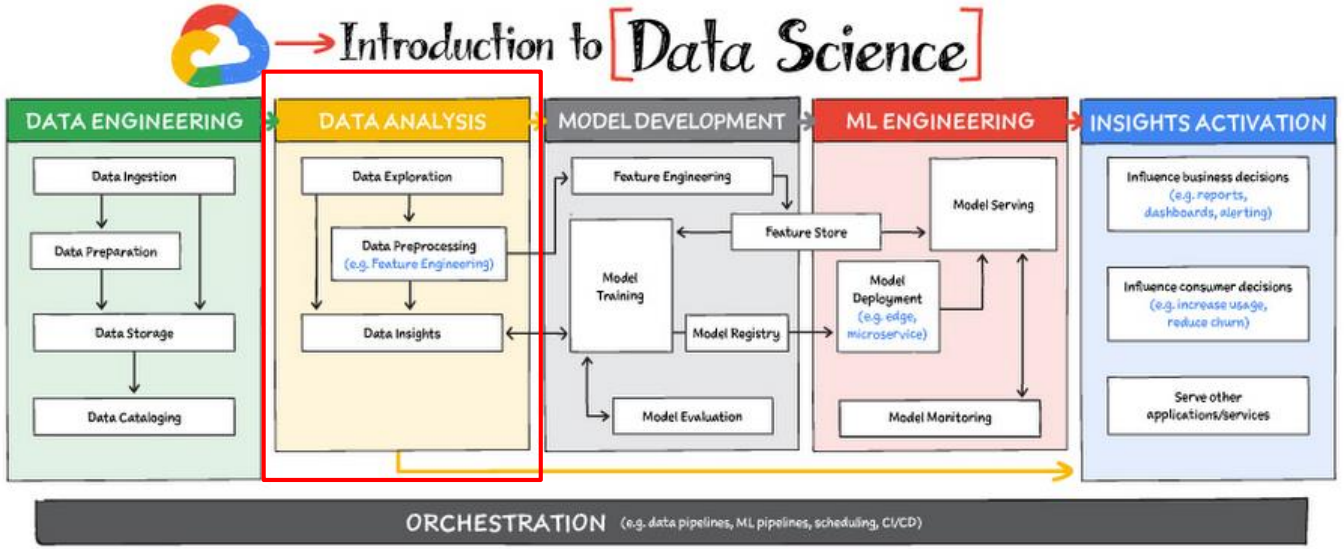
2. Get the Data - Create the workspace and download the data

- Work in an isolated environment may be prefer
  - Anaconda, virtualenv
- Get the data and convert the data to a format that you can manipulate
- Check the size and type of the data
- Create a test set and put it aside



3. Explore the data

- Create a copy of the data for exploration (sampling it down to a manageable size if needed)
- Visualizing the data
  - Study each attributes and its characteristics (exploratory data analysis and data cleaning)
  - Looking for correlations (feature selection)
  - Identify promising transformation (feature engineering)

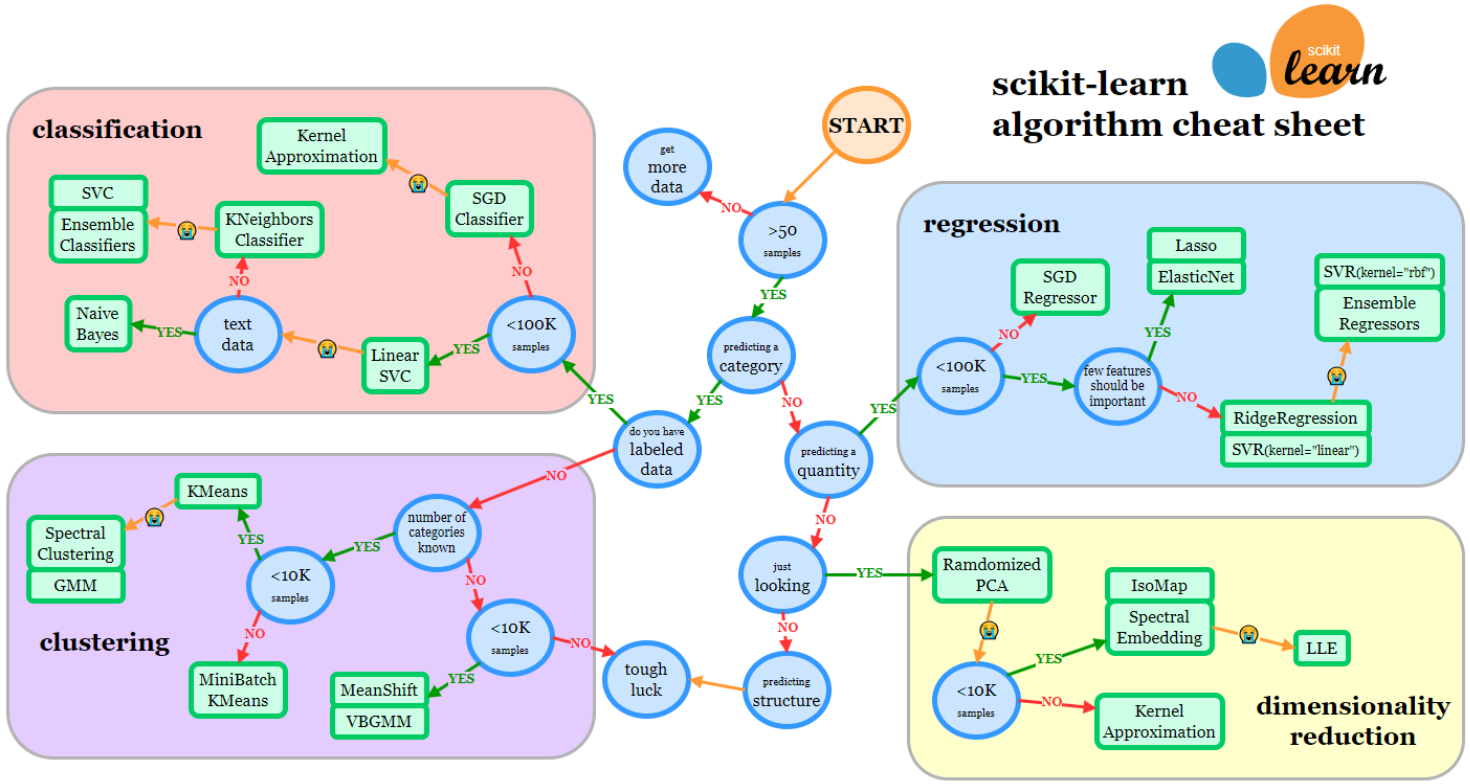


4. Prepare the data

- Data cleaning
  - Fixing outlier
  - Deal with missing data
- Feature selection
- Feature engineering
  - Handling text and categorical attributes
  - Decomposing features (date/time)
  - Aggregate features into promising ones
  - Add promising transforms of features
  - Discretize continuous feature
  - Feature scaling

5. Select a model and train it

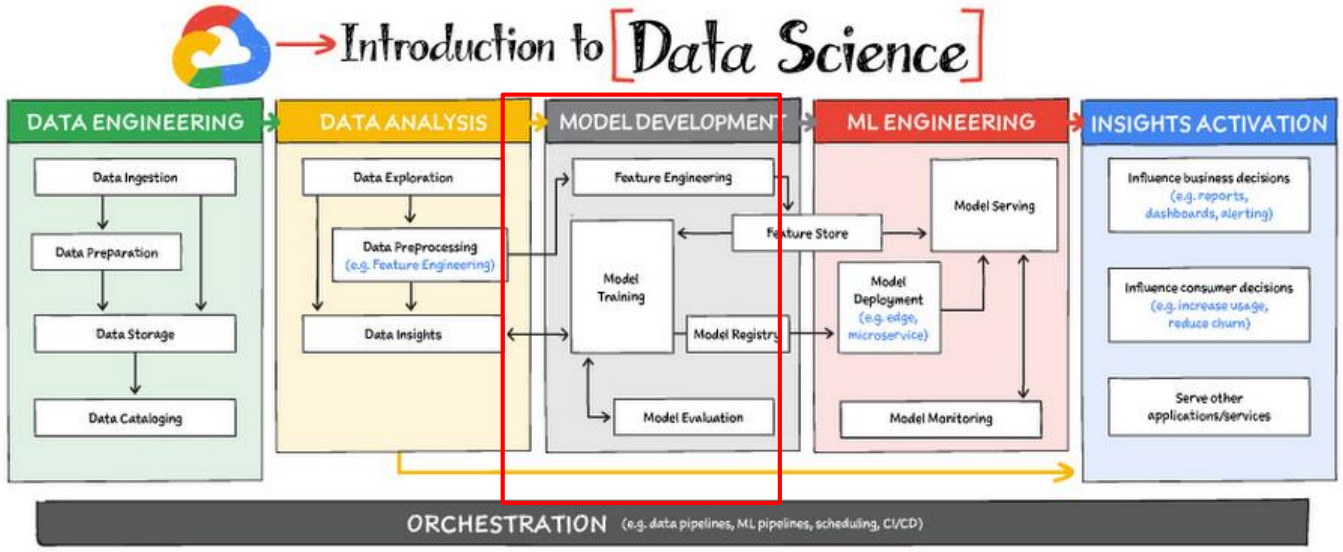
- Selecting models
  1. Training and evaluating on the training set
  2. Try different models
  3. Analyze the most significant variables for each of them
  4. Analyze the types of errors
  5. Shortlist promising models, preferring models that make different types of errors



https://scikit-learn.org/stable/machine_learning_map.html#choosing-the-right-estimator

6. Fine-Tune Your Model

- Fine-tune the hyperparameter using Cross-Validation
  - Grid search
  - Random search
  - You may want to try Bayesian optimization
- You can treat your data transformation choices as hyperparameters
- Try ensemble methods
- Measure performance on test set

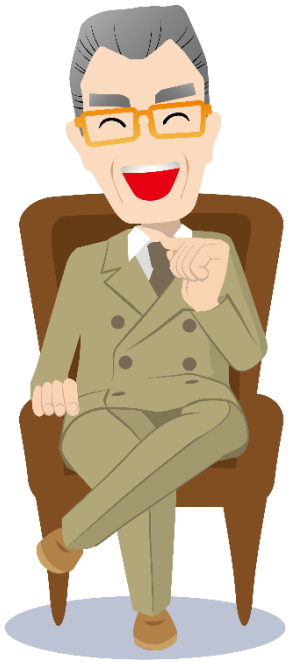


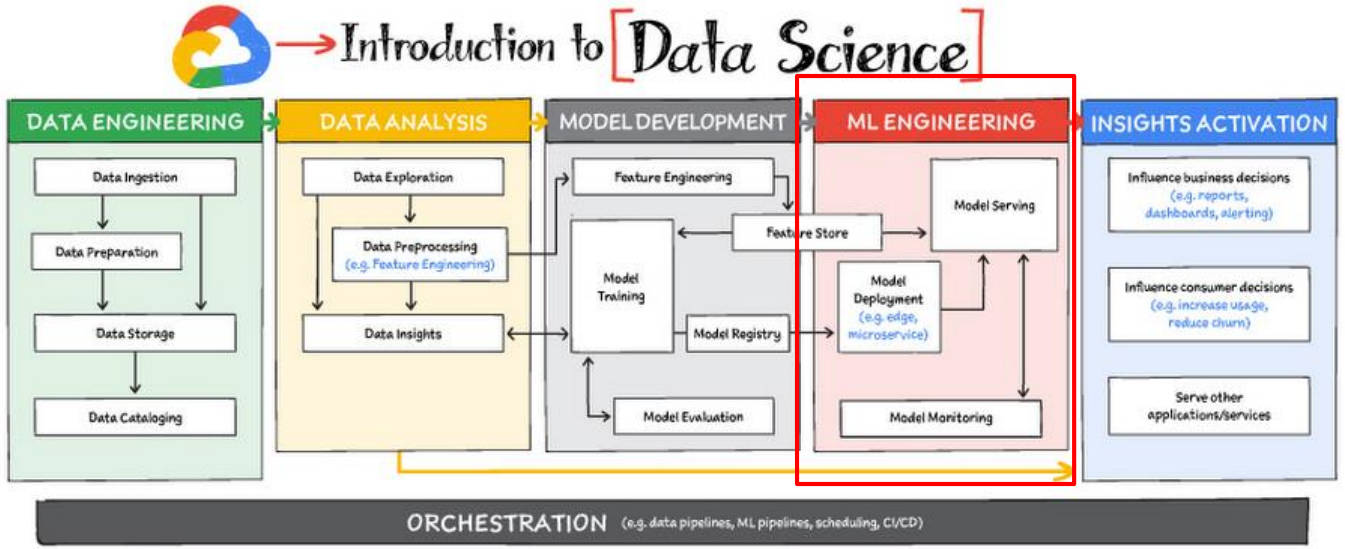
7. Present your solution

1. Document what you have done (Hackmd, Notebook, Notion)
2. Create a nice presentation
    1. Make sure you highlight the big picture first
3. Explain why your solution achieves the business objective
4. Don’t forget to present interesting points you noticed along the way
    1. Describe what worked and what did not
    2. List your assumptions and your system’s limitations
5. Ensure your key findings are communicated through beautiful visualizations or easy-to-remember statements (e.g., “the median income is the number-one predictor of housing prices”)

8. Launch, Monitor, and Maintain Your System

- Write monitoring code to check your system's live performance
  - Monitor input's quality
  - Retrain your model on a regular basis





Appendix

Working with Real Data

- Popular open data repositories
  - UC Irvine Machine Learning Repository
  - Kaggle datasets
  - Amazon's AWS datasets
  - Paperwithcode
- Meta portals (they list open data repositories)
  - OpenDataMonitor
  - Awesome datasets
- Other pages listing many popular open data repositories
  - Wikipedia's list of Machine Learning datasets
  - Quora.com
  - The datasets subreddit

Resources

- Data science
  - https://virgili0.github.io/Virgilio/#table-of-contents
  - https://e2eml.school/blog.html
  - https://github.com/GokuMohandas/MadeWithML
- Modeling
  - https://phonchi.github.io/nsysu-math524/
  - https://dafriedman97.github.io/mlbook/content/introduction.html
  - MOOC Lectures
- Deep learning
  - https://github.com/fastai/fastbook2e
  - https://d2l.ai/

Resources

- Libraries
  - https://github.com/EthicalML/awesome-production-machine-learning
  - https://github.com/ml-tooling/best-of-ml-python
  - https://github.com/krzjoa/awesome-python-data-science
- Cheat sheet
  - https://github.com/aaronwangy/Data-Science-Cheatsheet
  - https://www.mit.edu/~amidi/teaching/data-science-tools/
  - https://github.com/afshinea/stanford-cs-229-machine-learning
  - https://github.com/afshinea/stanford-cs-230-deep-learning

What is machine learning?

- What exactly does it mean for a machine to learn something?

- If I download a copy of Wikipedia, has my computer really learned something? Is it suddenly smarter?

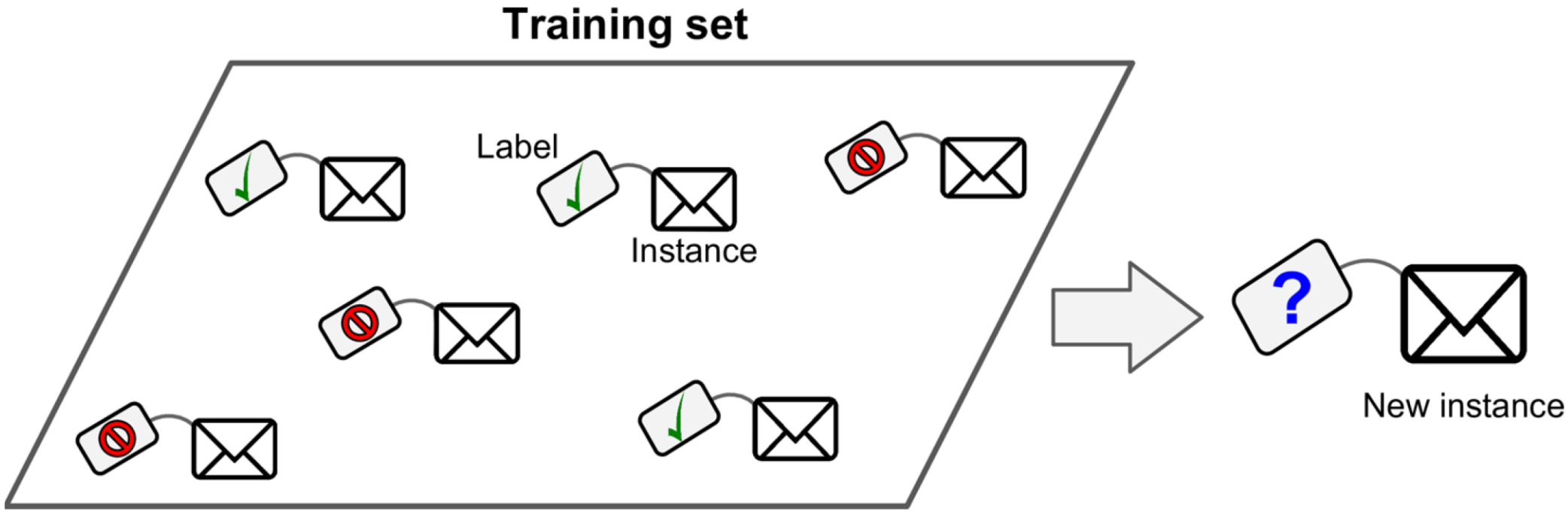
- Here is a slightly more general definition:

[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed. —Arthur Samuel, 1959

A computer program is said to learn from experience with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E. —Tom Mitchell, 1997

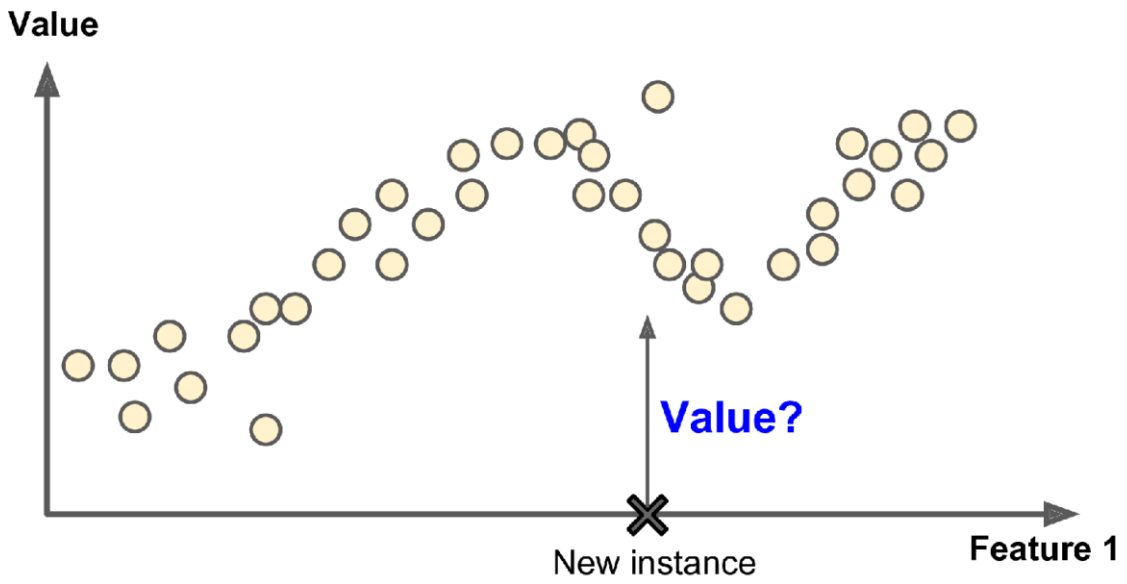
Types of Machine Learning Systems

- Supervised/Unsupervised Learning
  - In *supervised learning*, the training set you feed to the algorithm includes the desired solutions, called *labels*. Spam filter *is a classification task*



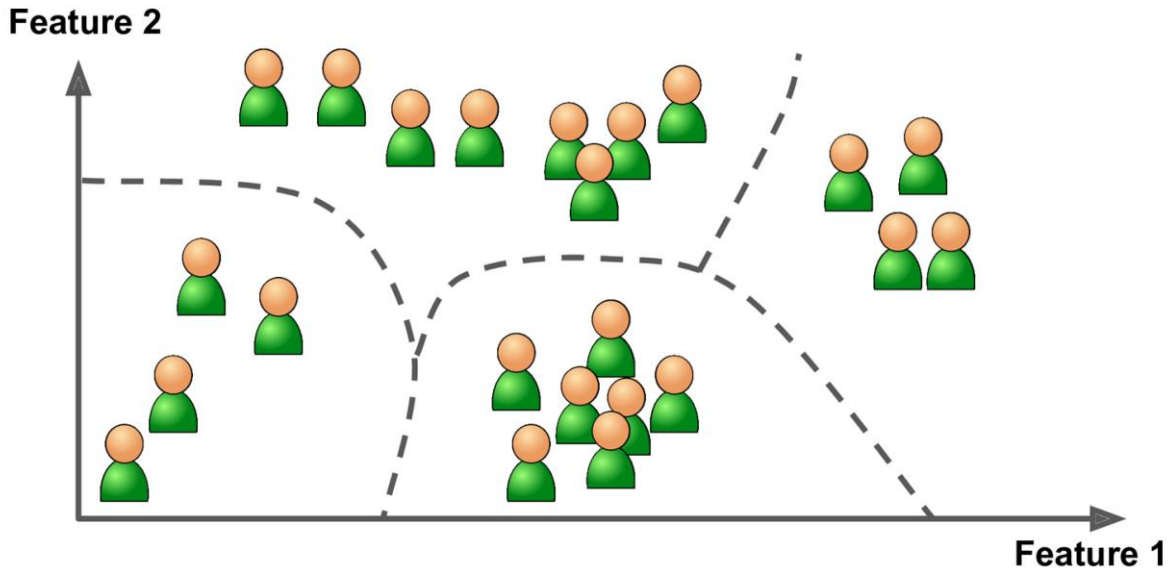
Types of Machine Learning Systems

- Supervised/Unsupervised Learning
  - Another typical task is to predict a *target* numeric value, such as the price of a car, given a set of *features* called *predictors*. This sort of task is called *regression*
  - Note that some regression algorithms can be used for classification as well, and vice versa. For example, *Logistic Regression* is commonly used for classification, as it can output a value that corresponds to the probability



Types of Machine Learning Systems

- Supervised/Unsupervised Learning
  - In *unsupervised learning*, as you might guess, the training data is unlabeled
  - For example, say you have a lot of data about your blog's visitors. You may want to run a *clustering* algorithm to try to detect groups of similar

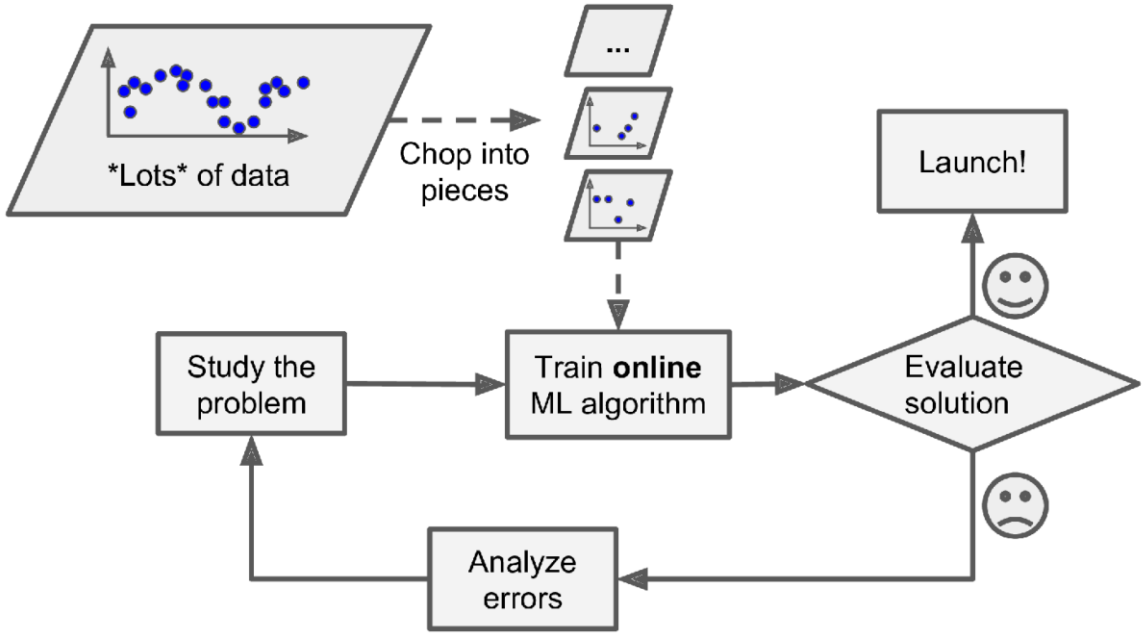


Types of Machine Learning Systems

- Batch and Online Learning
  - In *batch learning*, the system is trained using all the available data. This will generally take a lot of time and resources, so it is typically done offline. First the system is trained, and then it is launched into production and runs without learning anymore
  - If you want a batch learning system to know about new data (such as a new type of spam), you need to train a new version of the system from scratch on the full dataset, then stop the old system and replace it with the new one
    - It can be automated, so even a batch learning system can adapt to change. Simply update the data and train a new version of the system from scratch as often as needed
  - If your system needs to be able to learn autonomously and it has limited resources (e.g., a smartphone application or a rover on Mars), then carrying around large amounts of training data and taking up a lot of resources to train for hours every day is not good

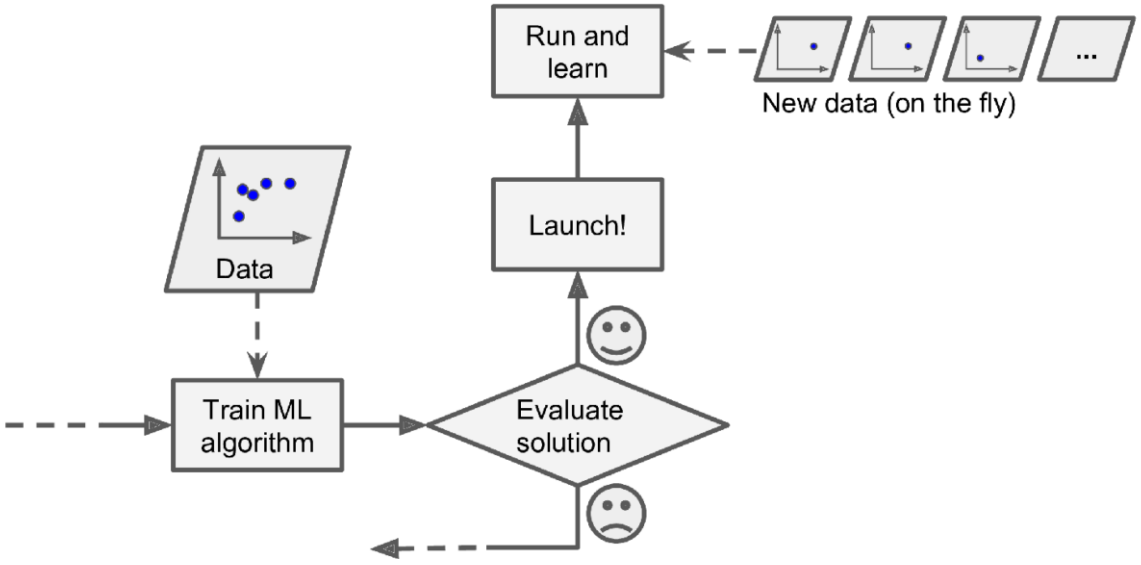
Types of Machine Learning Systems

- Batch and Online Learning
  - In *online learning*, you train the system incrementally by feeding it data sequentially, either individually or *mini-batches*. Each learning step is fast and cheap, so the system can learn about new data on the fly
  - It can receive data as a continuous flow and need to adapt to change rapidly
    - It is also a good option if you have limited computing resources
  - One important parameter of online learning systems is how fast they should adapt to changing data: this is called the *learning rate*



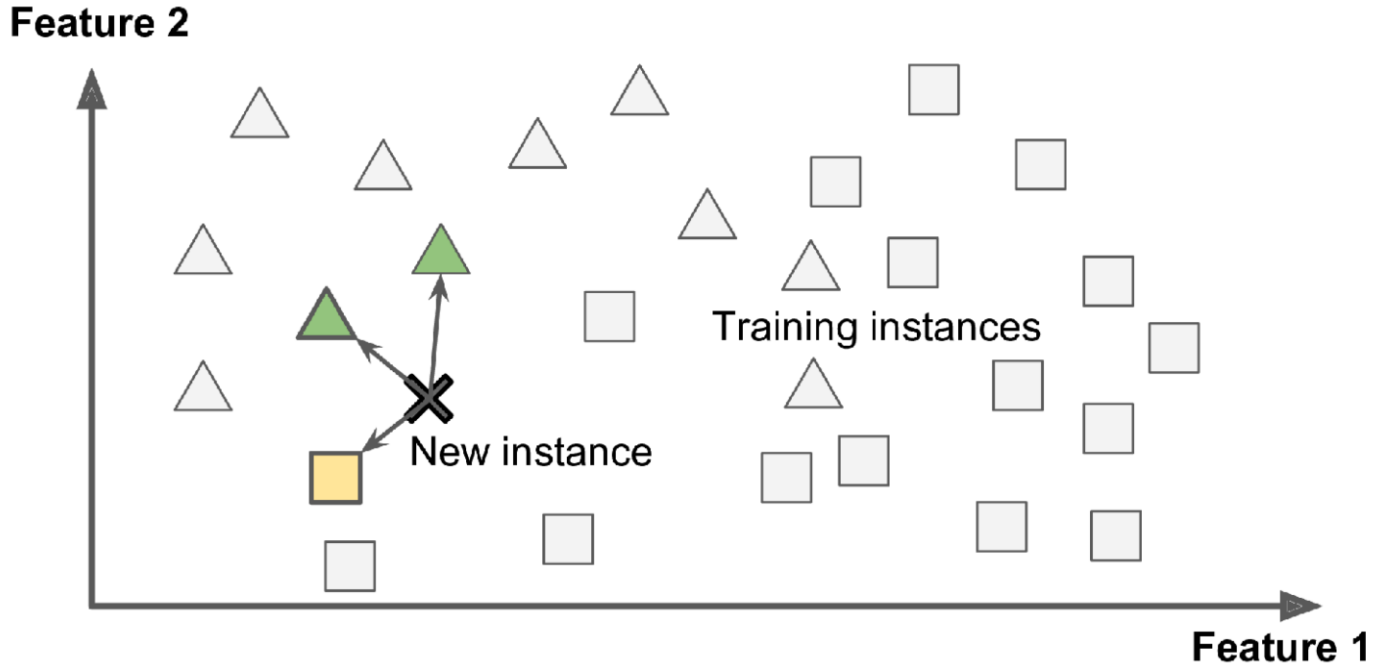
Types of Machine Learning Systems

- Batch and Online Learning
  - If bad data is fed to the system, the system's performance will gradually decline
  - For example, bad data could come from a malfunctioning sensor on a robot, or from someone spamming a search engine to try to rank high in search results. To reduce this risk you need to monitor your system closely and promptly switch learning off



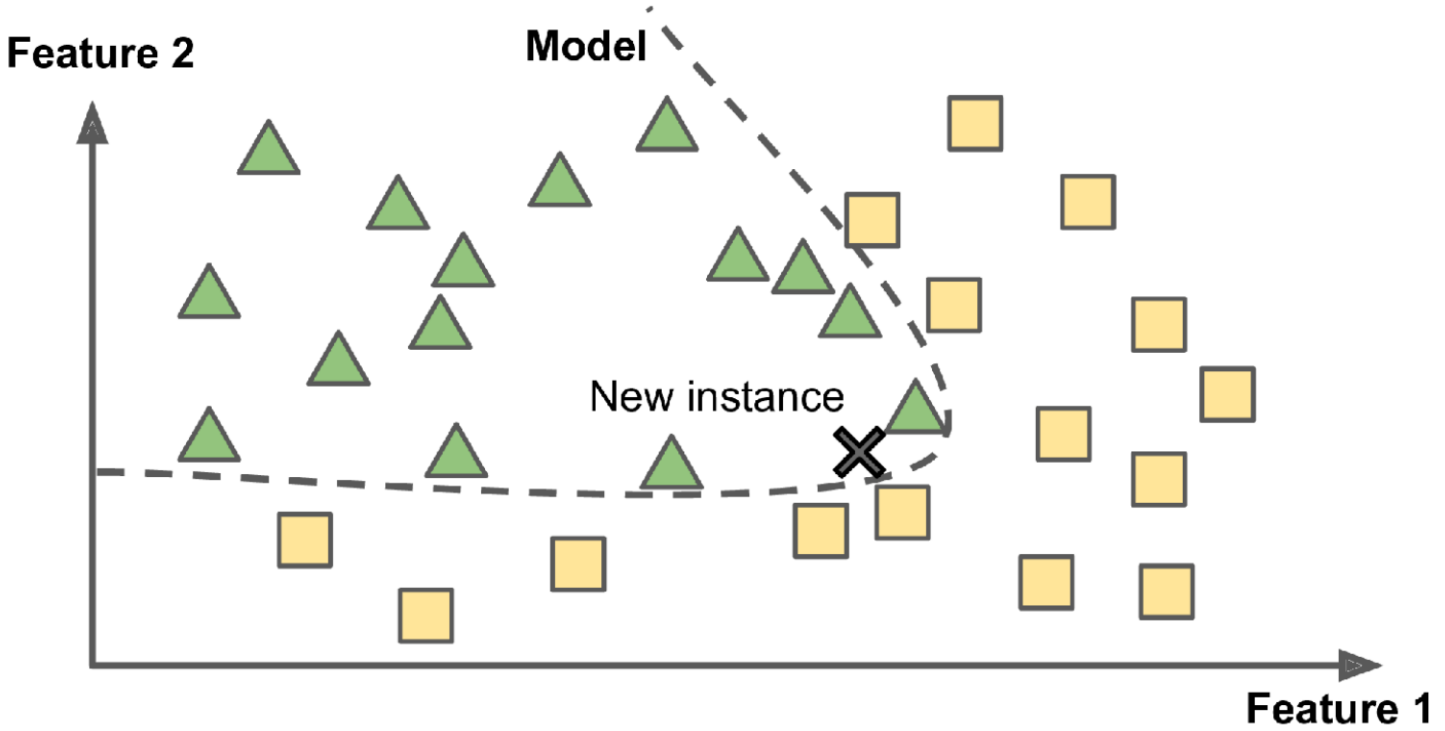
Types of Machine Learning Systems

- Instance-Based Versus Model-Based Learning
  - Spam filter could be programmed to flag emails that are similar to known spam. This requires a measure of similarity. It can count the number of words they have in common
  - The system would flag an email as spam if it has many words in common with a known spam email. This is called instance-based learning: the system learns the examples by heart, then generalizes to new cases using a similarity to compare them to the learned examples



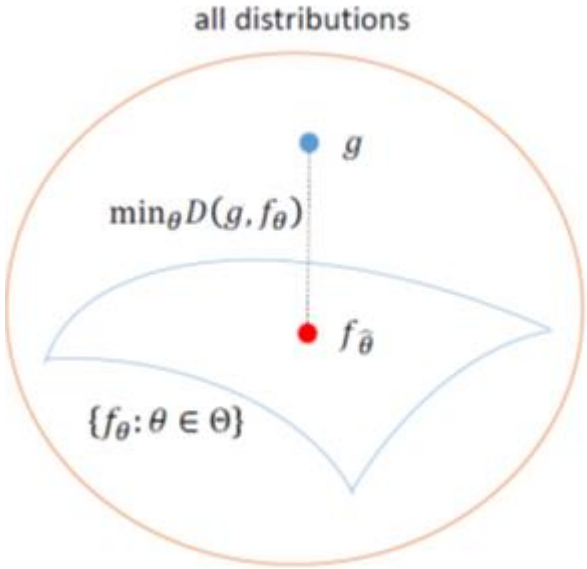
Types of Machine Learning Systems

- Instance-Based Versus Model-Based Learning
  - Another way to generalize from a set of examples is to build a model of these examples and then use that model to make predictions. This is called *model-based learning*



How to Train a Machine Learning Systems

- g is the distribution of data which is unknown
  - We have training set $\{(x_1, y_1), \dots, (x_n, y_n)\}$



1. Choose a <u>model</u> f_θ
   - Parametric
     - Explicit assumption and estimating a fix set of parameters
   - Non-parametric
     - No explicit assumption and need a large number of observations
2. Choose a quality measure (objective function, cost function) for fitting
   - Mean square error (Maximum likelihood)…
3. Optimization (fitting) to chose best θ
   - Calculus to find close form solution, gradient descent, expectation-maximization…

When to Use Machine Learning

- To summarize, Machine Learning is great for:
  1. Problems for which existing solutions require a lot of fine-tuning or long lists of rules: one Machine Learning algorithm can often simplify code and perform better than the traditional approach
  2. Complex problems for which using a traditional approach yields no good solution: the best Machine Learning techniques can perhaps find a solution
  3. Fluctuating environments: a Machine Learning system can adapt to new data
  4. Getting insights about complex problems and large amounts of data

Develop a model - Beat a baseline

- As you start working on the model itself, your initial goal is to achieve statistical power: that is, to develop a small model that is capable of beating a simple baseline. At this stage, these are the three most important things you should focus on:
  - Feature engineering—Filter out uninformative features (feature selection) and use your knowledge of the problem to develop new features that are likely to be useful
  - Selecting the correct architecture priors—What type of model architecture will you use? A densely connected network, a convnet, a recurrent neural network, a Transformer? Is deep learning even a good approach for the task, or should you use something else?
  - Selecting a good-enough training configuration—What loss function should you use? What batch size and learning rate?

Develop a model - Beat a baseline

- For most problems, there are existing templates you can start from. You're not the first person to try to build a spam detector, a music recommendation engine, or an image classifier
  - Make sure you research prior art to identify the feature engineering techniques and model architectures that are most likely to perform well on your task.
  - Note that it's not always possible to achieve statistical power. If you can't beat a simple baseline after trying multiple reasonable architectures, it may be that the answer to the question you're asking isn't present in the input data. Remember that you're making two hypotheses:
    - You hypothesize that your outputs can be predicted given your inputs
    - You hypothesize that the available data is sufficiently informative to learn the relationship between inputs and outputs
  - It may well be that these hypotheses are false, in which case you must go back to the drawing board

Develop a model

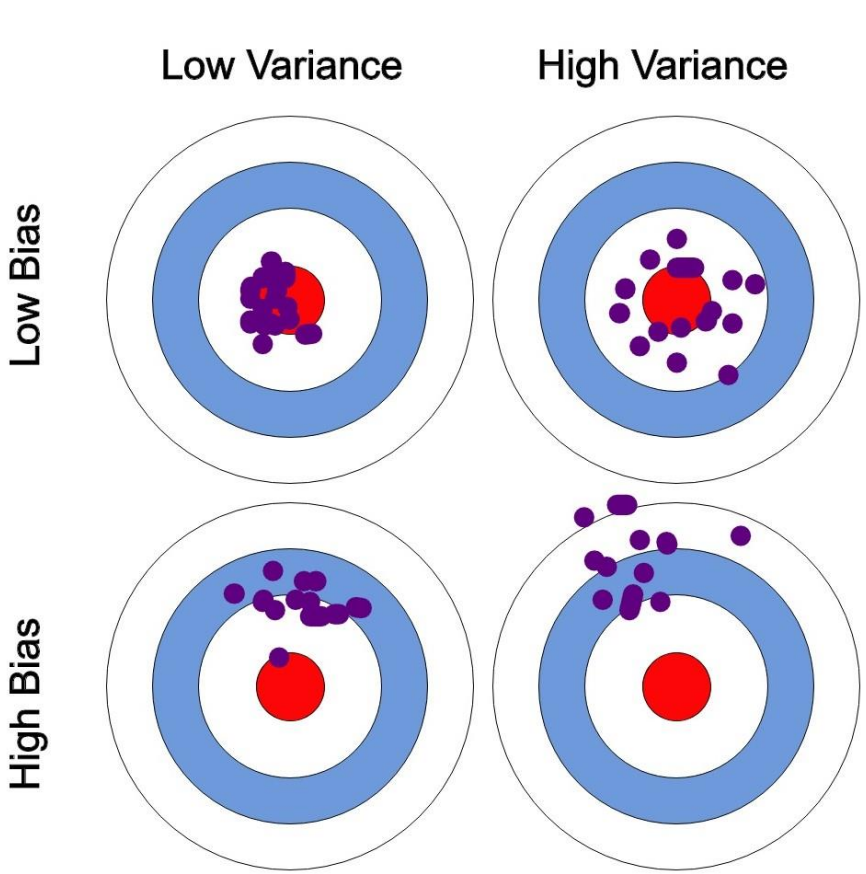
- Once you've obtained a model that has statistical power, the question becomes, is your model sufficiently powerful?
  - Remember that the universal tension in machine learning is between optimization and generalization. The ideal model is one that stands right at the border between underfitting and overfitting, between undercapacity and overcapacity. To figure out where this border lies, first you must cross it
  - To figure out how big a model you'll need, you must develop a model that overfits
- Always monitor the training loss and validation loss, as well as the training and validation values for any metrics you care about. When you see that the model's performance on the validation data begins to degrade, you've achieved overfitting

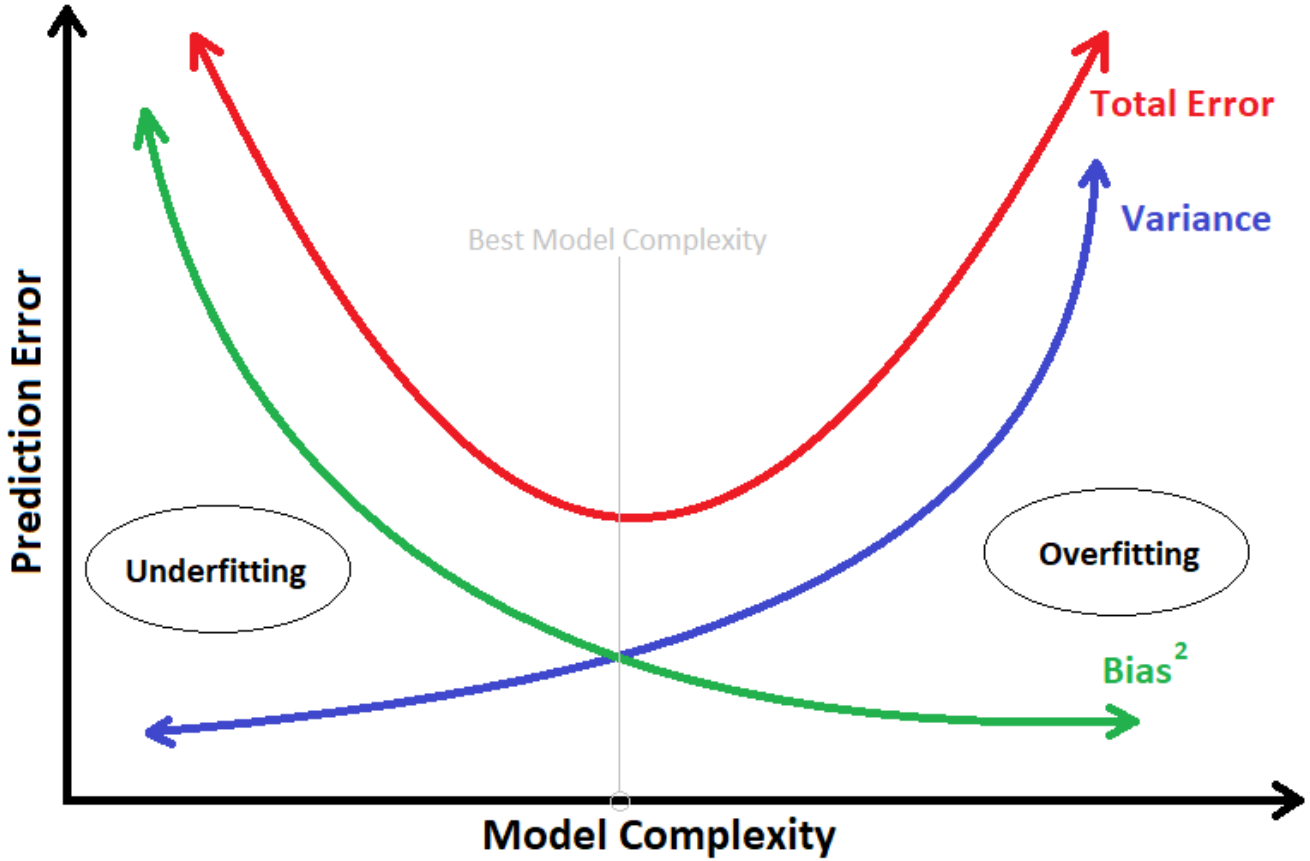
Develop a model - Some trade-offs

- No free lunch theorem
- A model is a simplified version of the observations. The simplifications are meant to discard the superfluous details that are unlikely to generalize to new instances. To decide what data to discard and what data to keep, you must make assumptions
  - If you make no assumption about the data, then there is no reason to prefer one model over any other. This is called the No Free Lunch (NFL) theorem. For some datasets the best model is a linear model, while for other datasets it is a neural network. The only way to know for sure which model is best is to evaluate them all

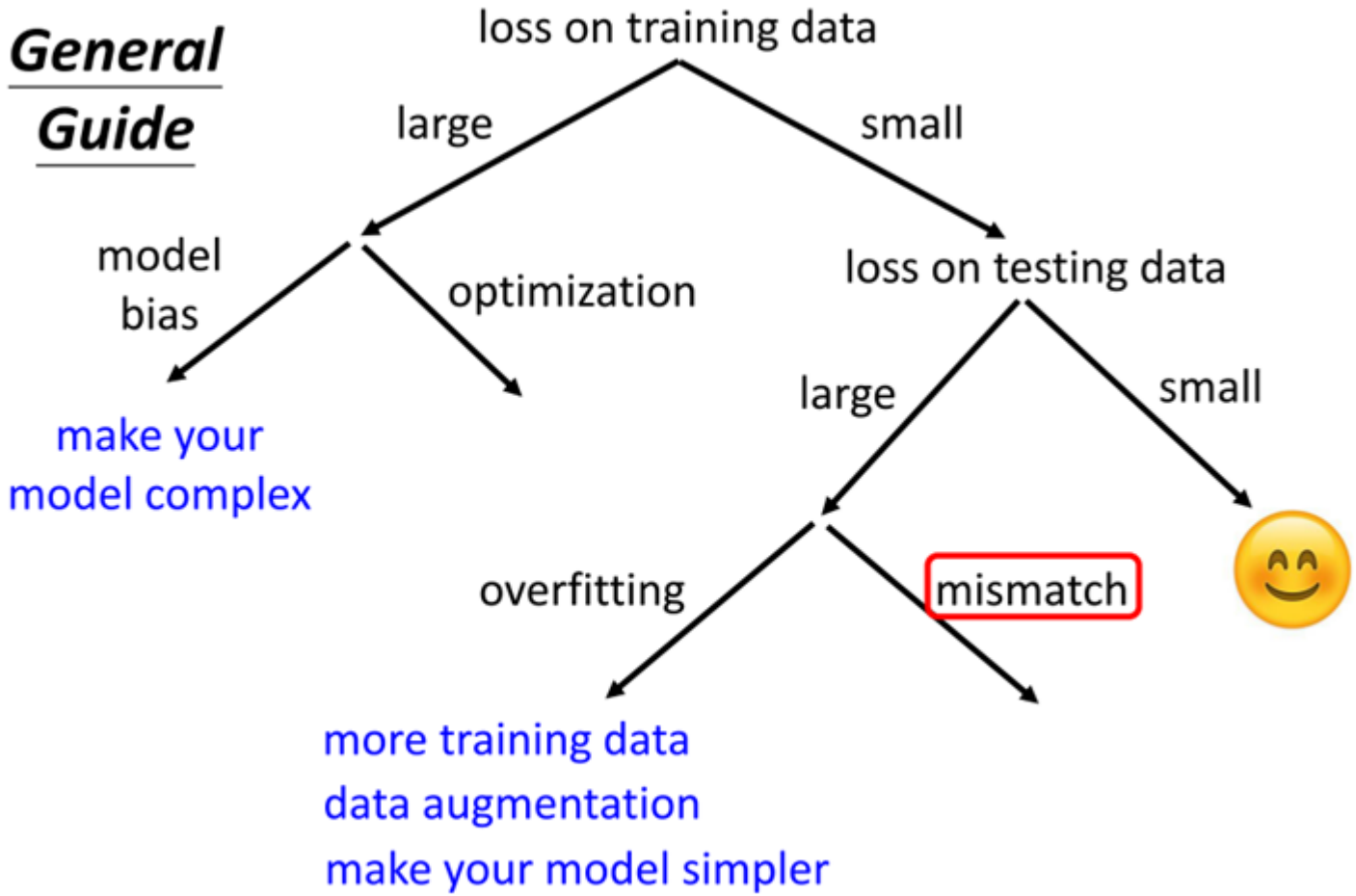
Develop a model - Some trade-offs

- Bias-variance trade-off



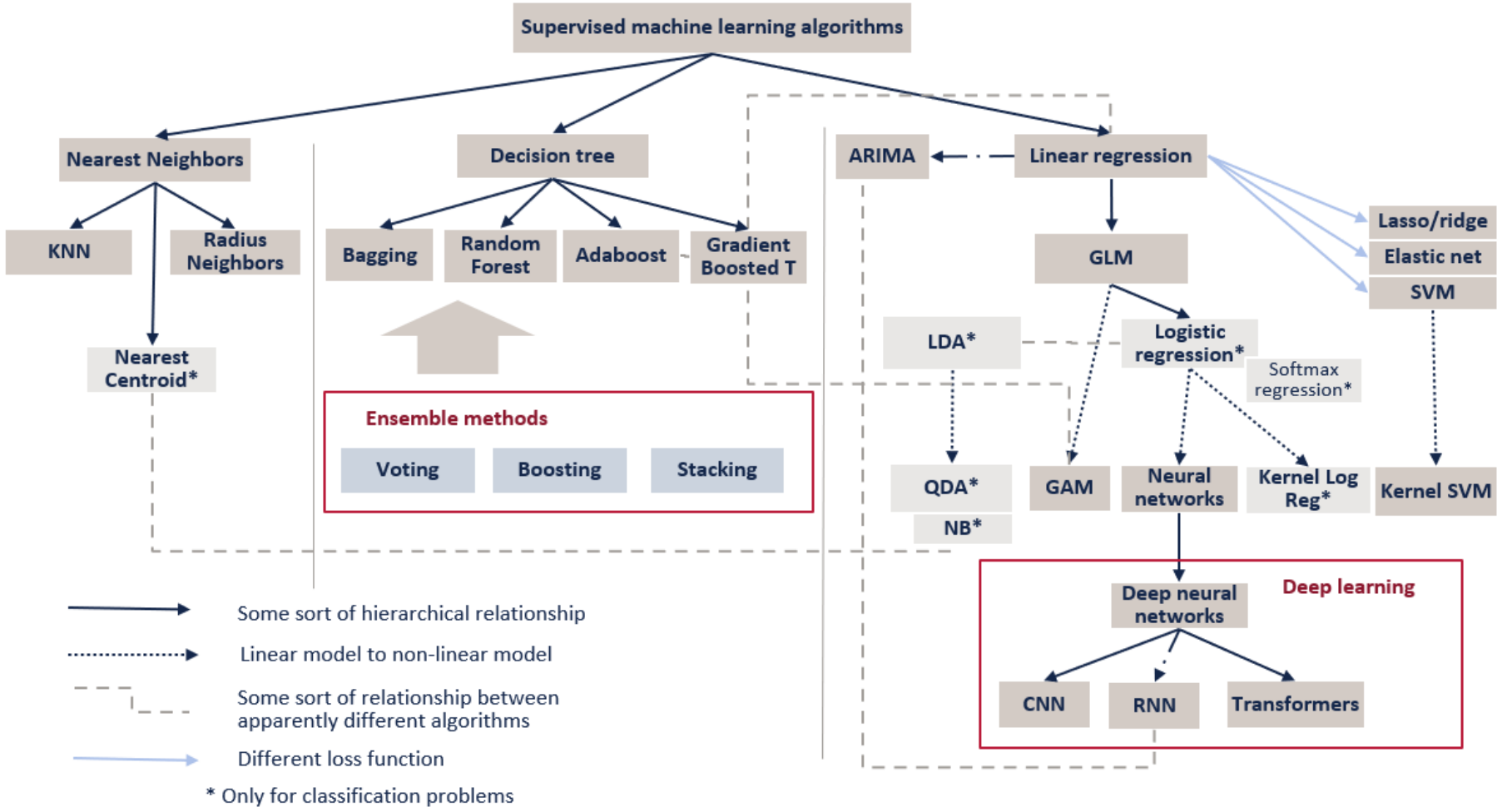


https://jason-chen-1992.weebly.com/home/-bias-variance-tradeoff



https://speech.ee.ntu.edu.tw/~hylee/ml/ml2021-course-data/overfit-v6.pdf

Develop a model - Supervised learning



https://towardsdatascience.com/overview-of-supervised-machine-learning-algorithms-a5107d036296