Data Science for Civil Engineering

Introduction

Jintao Ke

Department of Civil Engineering
Faculty of Engineering
The University of Hong Kong, Pokfulam, Hong Kong
Email: kejintao@hku.hk

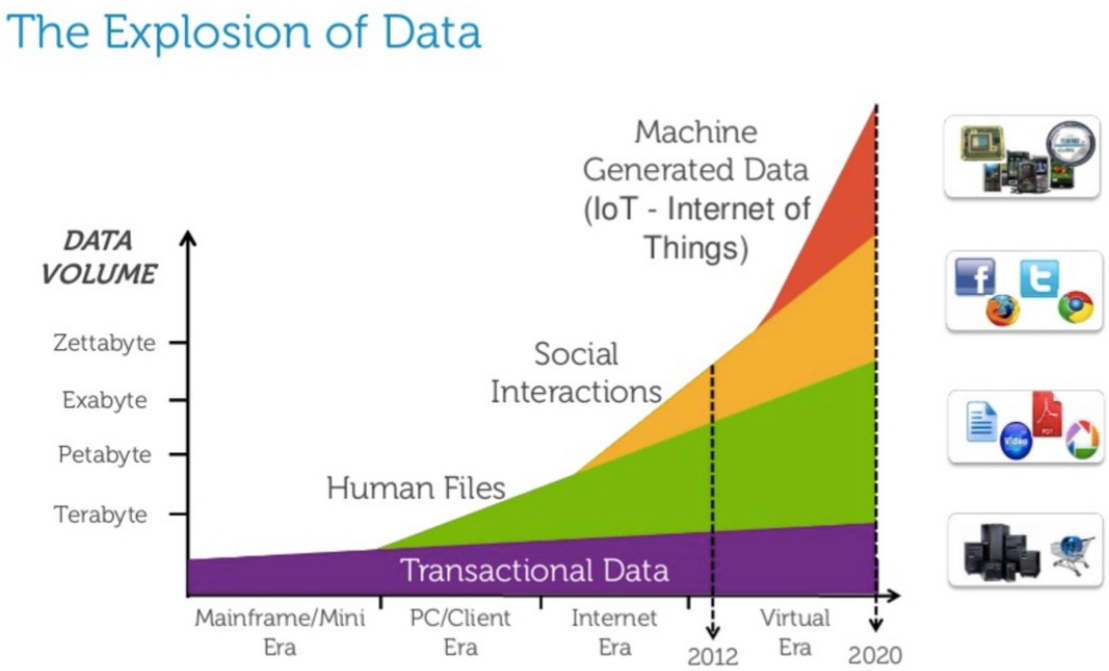
notes software: kami

Outline

Why We Need Data Science? I

The era of big data

- Massive amount of data is generated by human through transactions and social interactions and machines through the internet of things. In 2020, every person generated 1.7 megabytes in just a second.



Why We Need Data Science? II

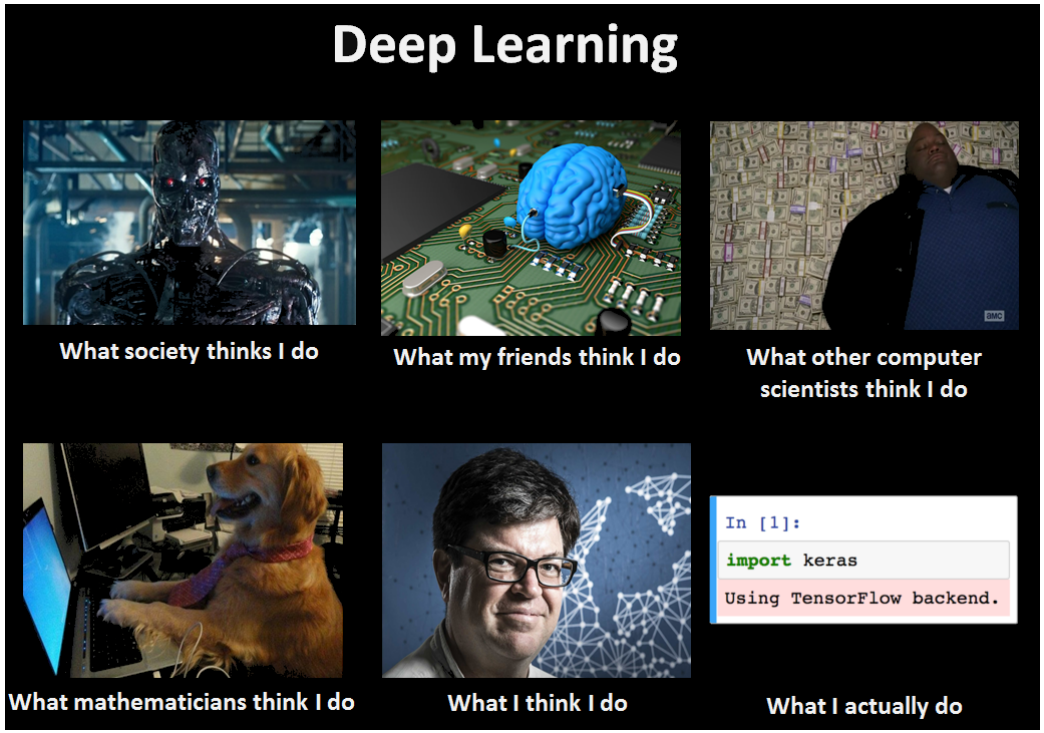
Why Data Science becomes so popular?

- Data science can help people make decisions in many fields, including marketing, social media, finance, healthcare, manufacturing as well as transportation.

What is Data Science? I

Data science is related to

- Classical statistics, Bayesian statistics and econometrics
- Machine learning, deep learning, artificial intelligence and data mining



What is Data Science? II

A formal definition

- Data Science is a blend of various tools, algorithms, and machine learning principles with the goal to discover hidden patterns from the raw data.
- Data Science is primarily used to make decisions and predictions by predictive causal analytics, prescriptive analytics (predictive plus decision science) and machine learning.

What is Data Science? III

Data science is used for

- Predictive causal analytics: Build a model to predict the possibilities of a particular event in the future.
- Prescriptive analytics: Not only predict but suggest a range of prescribed actions to achieve some desired outcomes (e.g., Google's self-driving car). The data gathered by vehicles can be used to train self-driving cars, such that the car make sequential decisions like when to turn, which path to take, when to slow down or speed up.

What is Machine Learning?

- Artificial Intelligence (AI) is intelligence demonstrated by machines, as opposed to natural intelligence displayed by animals including humans.
- Machine Learning is study of computer algorithms that can improve automatically through experience and by the use of data. It is regarded as a part of AI
- Deep Learning is a broader family of machine learning methods based on artificial neural networks with deeper structure and stronger ability to learn from big data
- Data Mining is the application of machine learning methods to large databases.

Applications of Machine Learning

Some emblematic fields of applications may involve:

- Machine Translation
- Image Generation
- Virtual Assistant
- Expert System
- Content Recommendation
- Poetry Writing
- Autonomous Vehicle
- Language Model
- Other Applications

Machine Translation

- Machine translation research started in the 1960s (the US government was quite keen on translating Russian into English). Over the subsequent decades, it went through quite a few rough turns.
- In the 1990s and 2000s, statistical machine translation, aided by large amounts of example human translations, helped vastly improve translation quality.

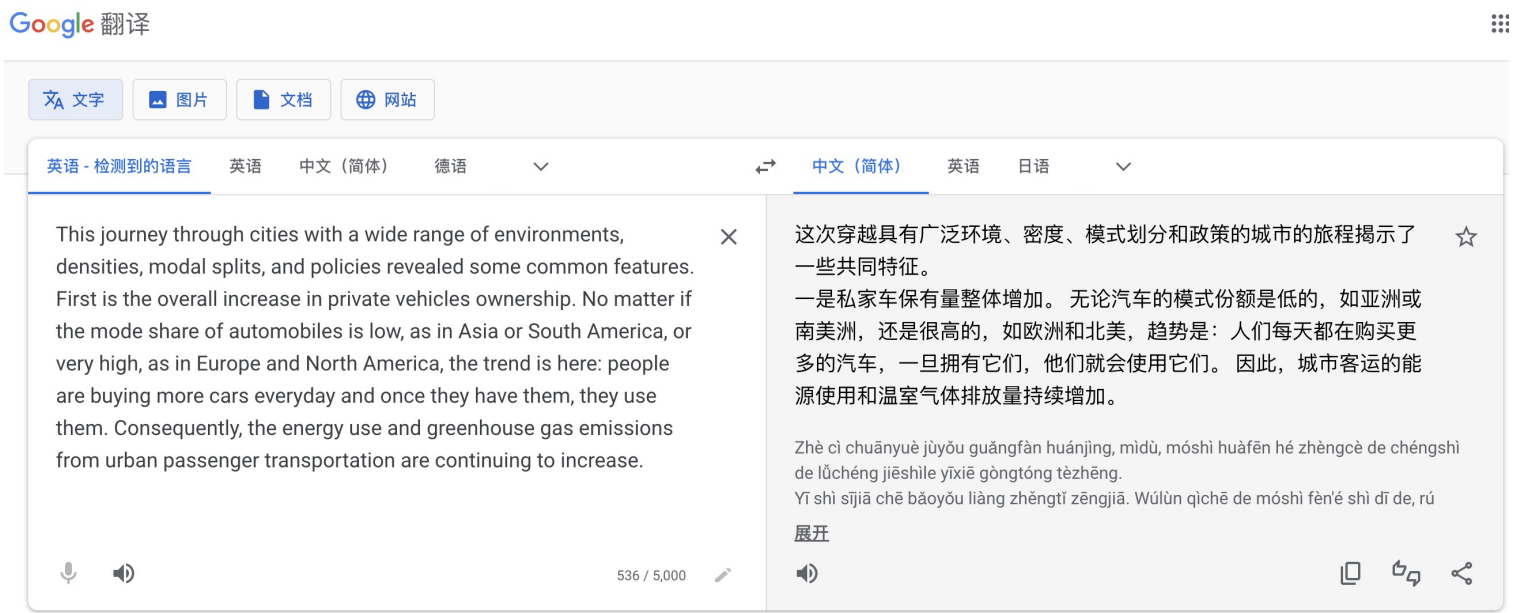


Image Generation

- One particular hot topic in computer vision is generating photorealistic images from text

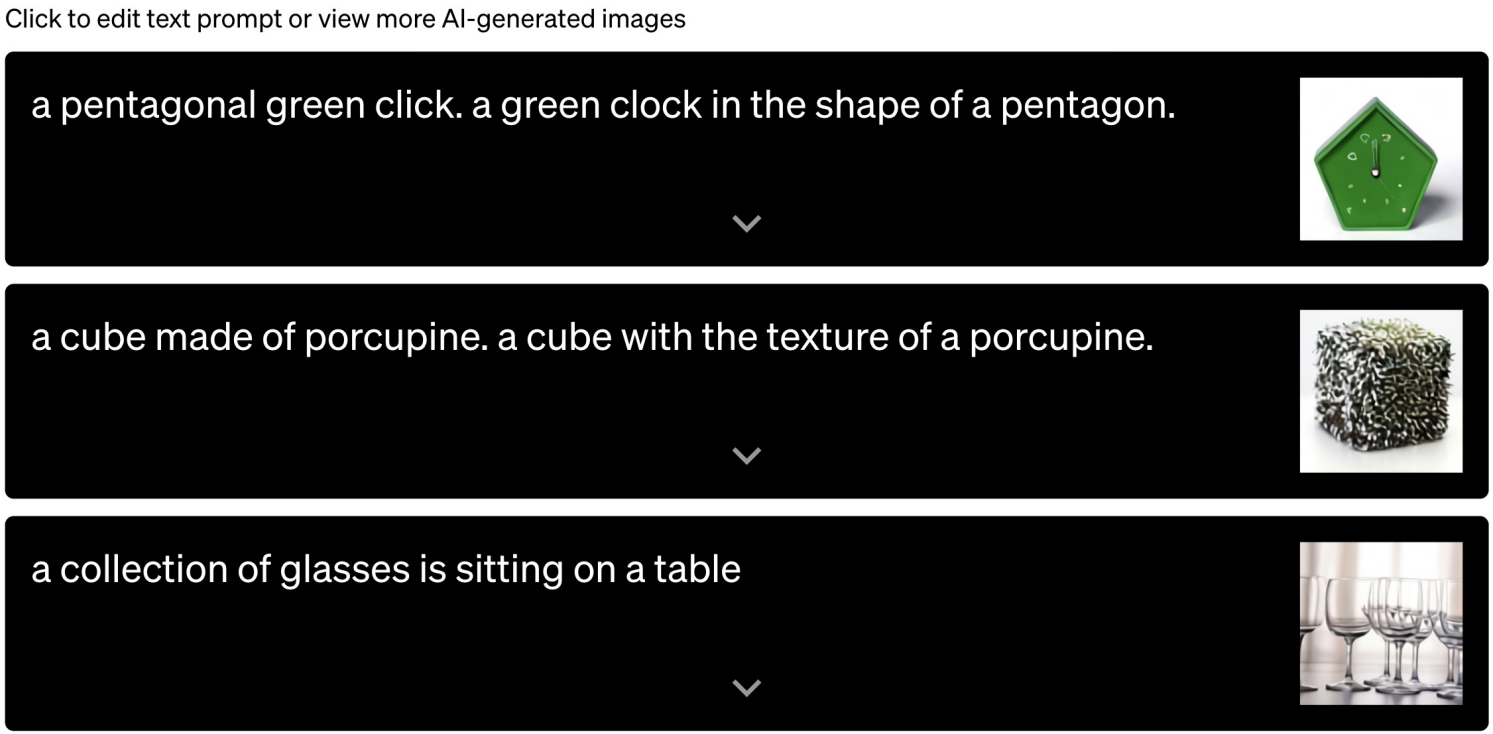


Figure: OpenAI-DALL.E: creating image from text

Virtual Assistant

- Speech recognition
- Text understanding

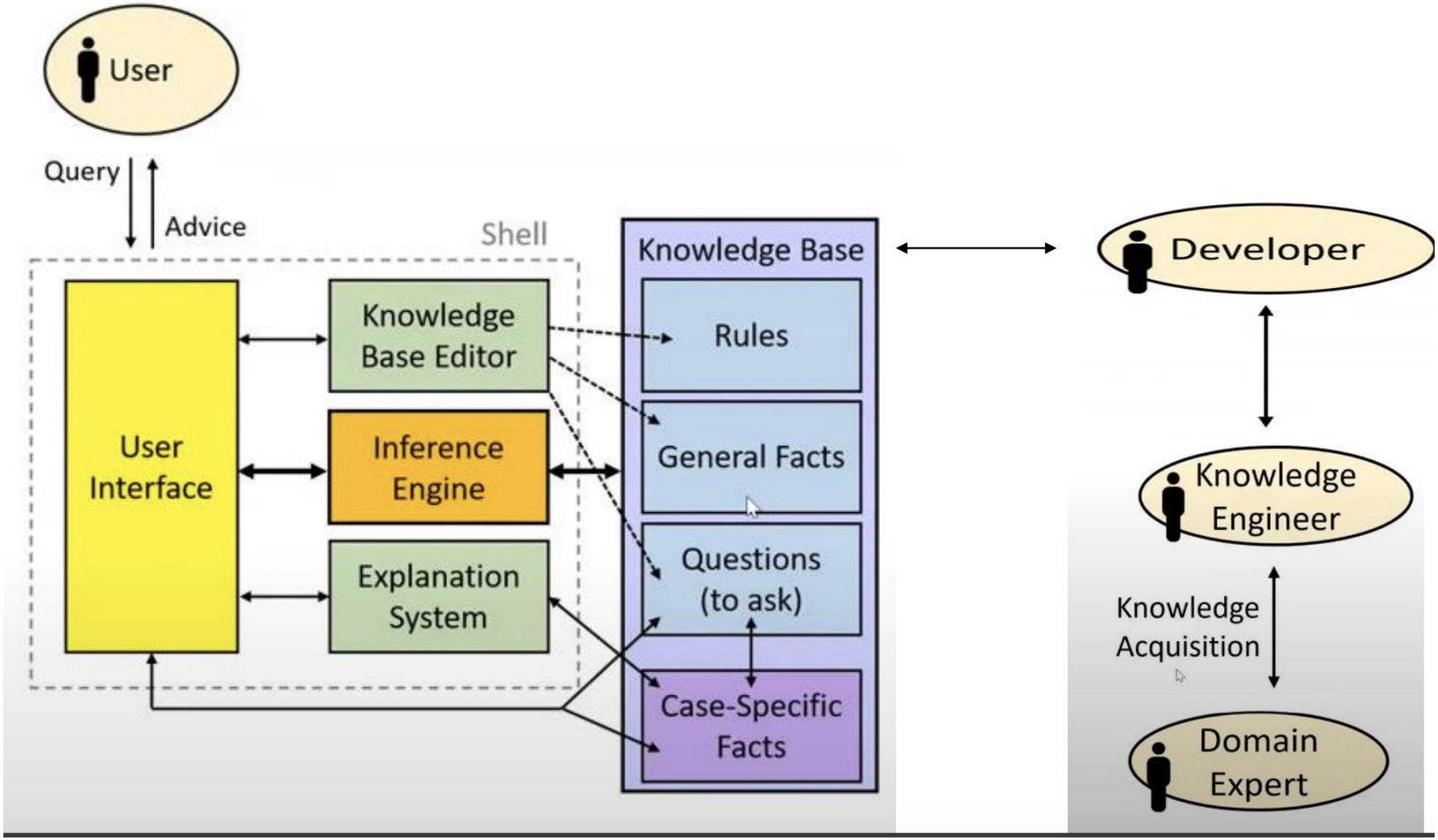
Figure: Virtual Assistant

Expert System

专家系统— 决定提什么问题， 怎么提问题，如何分析答案

- An Expert System (ES) is an example of a knowledge-based system. It was introduced around 1965 at Stanford by Edward Feigenbaum.
- "ES is a computer program that uses artificial intelligence (AI) technologies to simulate the judgment and behavior of a human or an organization that has expert knowledge and experience in a particular field such as medical diagnosis, account, coding, and games, etc."
- ES gathers data by asking the user questions about the problem. An initial set of questions can lead to further questions depending on the user's responses.
- The ES reasons what questions it needs to ask, based on the knowledge it is given. It will use the responses from users to rule out various possibilities that will allow it eventually reach a decision or diagnosis.

The General Design of an Expert System



Content Recommendation

- News (Yahoo)
- Post (Twitter, Instagram)
- Video (Youtube)
- Movie (Netflix)
- Music (Spotify)
- Shopping (Amazon)

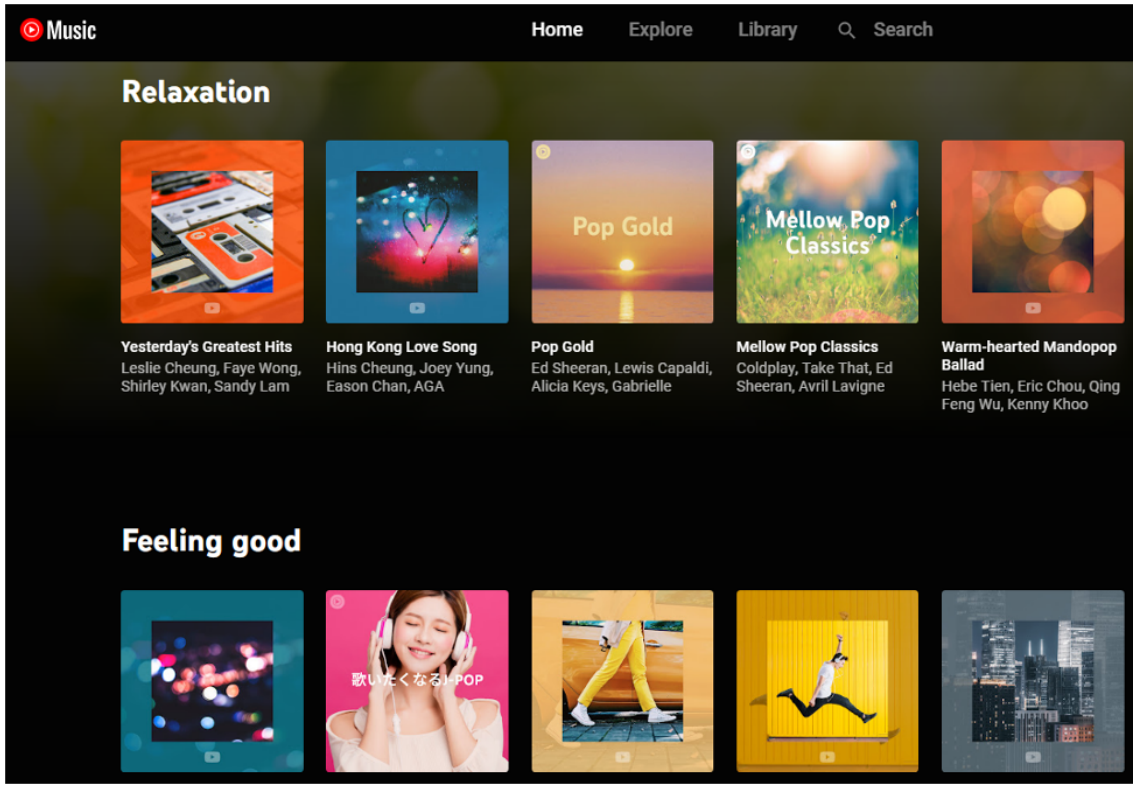


Figure: YouTube Music

Poetry Writing

- Verse by Verse is an experimental AI-powered muse that helps you compose poetry inspired by classic American poets

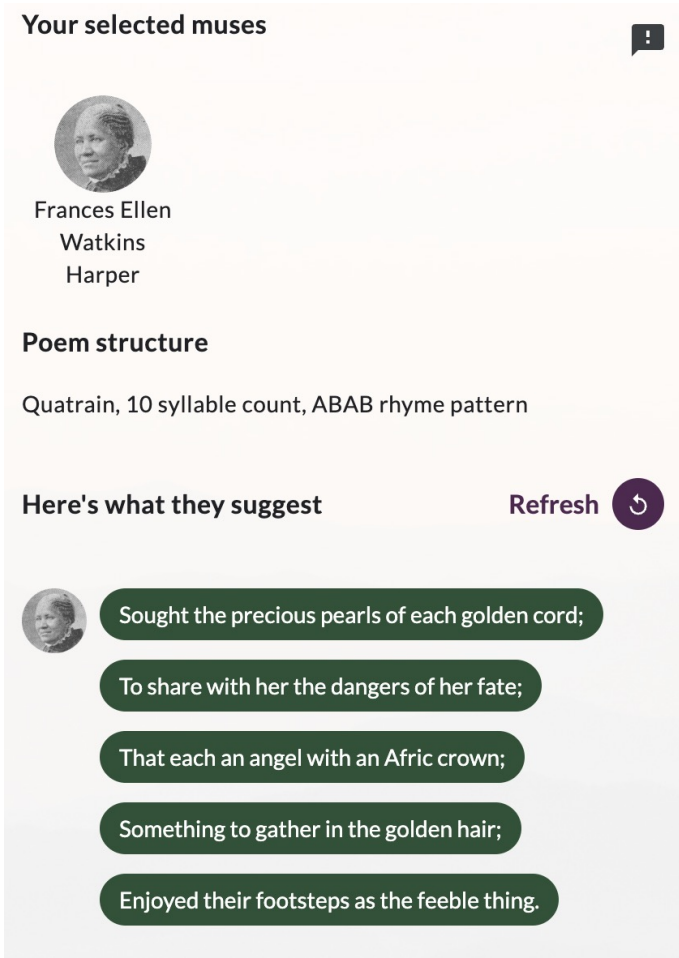


Figure: Google AI: Verse by Verse

Autonomous Vehicle

- Environment sensing
- Image understanding
- Route selection
- Steering
- Lane keeping/changing

Figure: Autonomous Drving

Categories of Machine Learning

Machine learning

- Supervised learning: Learn a general rule that maps inputs (features) to outputs (label) for making predictions.
  - Classification
  - Regression
- Unsupervised learning: Discover some hidden patterns from the input data itself. No label is given for learning
- Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal
- Supervised and unsupervised learning is more related to predictive causal analytics and can also support prescriptive analytics, while reinforcement learning directly provide solutions to prescriptive analytics

Data Samples I

Features and labels

- Features (Independent variables): we have a set of N samples $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, where $\mathbf{x}_n, \forall n \in 1, 2, ..., N$ is a vector with length D, indicating there is D features in each sample. The nth sample $\mathbf{x}_n = [x_{n1}, x_{n2}, ..., x_{nD}]$, while all samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]^T$
- Labels (Dependent variables): for each sample $n \in 1, 2, ..., N$, we have a target value or label y_n associated with the sample vector $\mathbf{x}_n$. All samples $\mathbf{Y} = [y_1, y_2, ..., y_n]^T$
- As for supervised learning, the objective is to find out a function $g(\mathbf{x})$ that takes a sample of features $\mathbf{x}_n$ ($\mathbf{x}$ for short) as input and generates an output $\hat{y} = g(\mathbf{x})$ for predicting the corresponding label y. The function $g(\mathbf{x})$ can also be expressed as $g(\mathbf{x}|\mathbf{w})$ parameterized by a set of parameters $\mathbf{w}$.

Data Samples II

在机器学习中，**泛化（Generalization）**是指模型对未见过的新数据的处理能力。

简而言之，一个模型的泛化能力就是它在新数据上的表现，这些新数据在模型的训练过程中是未被观察到的。

Training and Testing

- Training data (set): a sample of data with features and labels $(\mathbf{x}_n, y_n), \forall n \in 1, 2, ..., N$ available to learn parameters $\boldsymbol{w}$ that can help $g(\mathbf{x}|\mathbf{w})$ to predict y as accurately as possible.
- Test data (set): once the model is trained (i.e., parameters $\mathbf{w}$ are determined) on the basis of training data, the model can predict y based on the features of new samples, which are said to comprise a *test set*. The ability to correctly predict new samples' features is known as generalization.
- Validation set: a part of training data extracted to tune the hyperparameters (i.e. the architecture) of a model and used for early stopping in the training of neural networks.

泛化的重要性

一个理想的机器学习模型不仅能够很好地拟合其训练数据，也能够在新的、未见过的数据上表现得很好。如果一个模型在训练数据上表现很好，但在新数据上表现差，我们就说这个模型出现了**过拟合（Overfitting）**。相反，如果模型在训练数据上的表现不足，也无法捕捉到数据的基本关系，就称为**欠拟合（Underfitting）**。

Regression

为了提高模型的泛化能力，研究者和工程师会使用多种技术：

- **数据集划分**：将数据分为训练集、验证集和测试集，其中验证集用于模型选择和调整超参数，测试集用于评估模型的泛化能力。
- **交叉验证**：特别是在数据量不足的情况下，使用交叉验证可以更可靠地估计模型的泛化能力。
- **正则化**：如L1、L2正则化，可以减少模型复杂度，避免过拟合。
- **集成学习**：通过组合多个模型来提高泛化能力，例如随机森林或梯度提升机。

Training Phase

- A sample of data $\mathcal{X}$ (training data) with features and labels $(\mathbf{x}_n, y_n), \forall n \in 1, 2, ..., N$ are collected, where y is a continuous value (e.g., any real value in the range of $[-\infty, \infty]$).
- Use the training set to find out the set of parameters or weights $\hat{\mathbf{w}}$ that makes $g(\mathbf{x}|\mathbf{w})$ predict y as accurately as possible.

- **数据增强**：通过增加训练样本的多样性来提高模型的鲁棒性和泛化能力。
- **早停（Early Stopping）**：在训练过程中，一旦在验证集上的表现不再提高，就停止训练，以避免过拟合。
- **超参数调优**：通过搜索最优超参数配置来找到最佳的模型复杂度。

Testing Phase

- For any new testing samples with features $\mathbf{x}_n$, we can use the trained model to estimate the corresponding $\hat{y}_n = g(\mathbf{x}_n|\hat{\mathbf{w}})$, which is denoted as predicted label.
- The actual label y_n corresponding to the sample $\mathbf{x}_n$ is called the true label. Prediction errors can be measured by the difference between predicted label $\hat{y}_n$ and true label y_n.

Error Minimization

In training phase, how to find out the optimal parameters?

- This can be done by minimizing an error function that measures the misfit between the function $g(\mathbf{x}|\mathbf{w})$, for any given value of $\mathbf{w}$, and the true labels y. One simple error function is given by the sum of the squares of the errors between the predictions $g(\mathbf{x}_n|\mathbf{w})$ for each data point $\mathbf{x}_n$ and the corresponding true labels y_n

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}[\hat{y}_n - y_n]^2 = \frac{1}{2}\sum_{n=1}^{N}[g(\mathbf{x}_n|\mathbf{w}) - y_n]^2 \tag{1}$$

- where the factor of 1/2 is included for later convenience. So the optimal parameters can be obtained by

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} E(\mathbf{w}) \tag{2}$$

Performance Evaluation for Regression

对于分类任务：

准确度（Accuracy）：准确度是最直观的性能度量，它是分类正确的样本数除以总样本数。

混淆矩阵（Confusion Matrix）：混淆矩阵是一个更详细的性能描述，它显示了实际类别与模型预测类别的关系。

精确率（Precision）：精确率是模型预测为正类别中实际为正类别的比例。

召回率（Recall）或灵敏度（Sensitivity）：召回率是实际为正类别中模型预测为正类别的比例。

In testing phase, how to measure prediction accuracy?

Given N samples in the test set, use the model $g(\mathbf{x}_n|\hat{\mathbf{w}})$ to estimate $\hat{y}_n$ for each testing sample with features $\mathbf{x}_n$. Then the performance measures can be:

- Mean squared error (MSE): $\frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)^2$
- Root mean squared error (RMSE): $\sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)^2}$
- Mean absolute error (MAE): $\frac{1}{N}\sum_{n=1}^{N}|\hat{y}_n - y_n|$
- Coefficient of determination (R^2): $1 - \frac{\sum_{n=1}^{N}(\hat{y}_n - y_n)^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2}$, where $\bar{y}$ is the mean value of y_n in the test set. $R^2 = 1$ when predicted label values exactly match the observed true labels.

F1分数（F1 Score）：F1分数是精确率和召回率的调和平均数，它试图同时考虑精确率和召回率。

ROC曲线（Receiver Operating Characteristic Curve）：ROC曲线下面积（AUC）是另一种评价分类器性能的方法，它不依赖于特定的分类阈值。

PR曲线（Precision-Recall Curve）：PR曲线展示的是精确率和召回率的关系，对于不平衡类别的数据特别有用。

Classification I

- Training set with N samples: $\mathcal{X} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$
- Each sample (e.g., a car) contains a vector of features $\mathbf{x}_n$ and a label y_n which represents the class of the sample (e.g., family cars for positive samples or other cars for negative samples).
  - Feature vector: $\mathbf{x}_n = (x_{n1}, x_{n2})^T$ (only 2 features for simplicity, e.g., x_{n1} = price, x_{n2} = engine power)
  - Class label: $y_n = \begin{cases} 1, & \text{if } \mathbf{x}_n \text{ is a positive sample} \\ 0, & \text{if } \mathbf{x}_n \text{ is a negative sample} \end{cases}$
- Use $\mathcal{X}$ to train a decision rule to classify a sample $\mathbf{x}$ into either a positive or negative sample, based on the discriminant functions $g(\mathbf{x}_n|\mathbf{w})$, for example,

$$\hat{y}_n = \begin{cases} 1, & \text{if } g(\mathbf{x}_n|\mathbf{w}) > 0 \\ 0, & \text{if } g(\mathbf{x}_n|\mathbf{w}) < 0 \end{cases} \tag{3}$$

Classification II

Testing Phase

- For any new testing samples with features $\mathbf{x}_n$, we can use the trained model to estimate the discriminant function $g(\mathbf{x}_n|\mathbf{w})$.
- Given a threshold τ, we can classify a sample $\mathbf{x}_n$ into a positive sample, i.e., $\hat{y}_n = 1$, if $g(\mathbf{x}_n|\mathbf{w}) > \tau$, and into a negative sample , i.e., $\hat{y}_n = 0$, if $g(\mathbf{x}_n|\mathbf{w}) < \tau$.
- The actual label y_n corresponding to the sample $\mathbf{x}_n$ is called the true label, while $\hat{y}_n$ is the predicted label. Prediction errors can be measured by the difference between predicted label $\hat{y}_n$ and true label y_n.

Key question

- How to find out a discriminant function $g(\mathbf{x}_n|\mathbf{w})$ to perform optimal classification?

Maximum Likelihood Estimation

- Different classifiers (classification models) may have different forms of discriminant functions parameterized by $\mathbf{w}$ and different ways to solve parameters $\mathbf{w}$. One widely used approach is maximum likelihood estimation (MLE):
- MLE seeks estimates for $\mathbf{w}$ such that the predicted probability $P(y_n = 1|\mathbf{x}_n, \mathbf{w})$, $P(y_n = 0|\mathbf{x}_n, \mathbf{w})$ for each sample $\mathbf{x}_n$ corresponds as closely as possible to the sample's true label y_n. The likelihood of $\mathbf{w}$ given the training set $\mathcal{X}$:

$$L(\mathbf{w}|\mathcal{X}) = \prod_{n=1}^{N} \left[P(y_n = 1|\mathbf{x}_n, \mathbf{w})^{y_n} P(y_n = 0|\mathbf{x}_n, \mathbf{w})^{1-y_n} \right] \quad (4)$$

- when the true label $y_n = 1$, the term in the bracket is $P(y_n = 1|\mathbf{x}_n, \mathbf{w})$, when $y_n = 0$, the term in the bracket is $P(y_n = 0|\mathbf{x}_n, \mathbf{w})$.

Maximum Likelihood Estimation

- We let $g(\mathbf{x}_n|\mathbf{w}) = P(y_n = 1|\mathbf{x}_n, \mathbf{w})$ to represent the probability of $y_n = 1$ given a sample $\mathbf{x}_n$ and parameters $\mathbf{w}$, then $P(y_n = 0|\mathbf{x}_n, \mathbf{w}) = 1 - g(\mathbf{x}_n|\mathbf{w})$, and the likelihood function can be written as:

$$L(\mathbf{w}|\mathcal{X}) = \prod_{n=1}^{N} \left\{ g(\mathbf{x}_n|\mathbf{w})^{y_n} \left[1 - g(\mathbf{x}_n|\mathbf{w})\right]^{1-y_n} \right\} \tag{5}$$

- The optimal parameters $\hat{\mathbf{w}}$ can be obtained by maximizing the likelihood function $L(\mathbf{w}|\mathcal{X})$, namely,

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} L(\mathbf{w}|\mathcal{X}) \tag{6}$$

Maximum Likelihood Estimation

- As usual, we can define an error function by taking the negative logarithm of the likelihood, which gives the cross-entropy error function as follows:

$$\begin{aligned} E(\mathbf{w}|\mathcal{X}) &= -\log L(\mathbf{w}|\mathcal{X}) \\ &= -\sum_{n=1}^{N} \{y_n \log g(\mathbf{x}_n|\mathbf{w}) + (1-y_n)\log[1-g(\mathbf{x}_n|\mathbf{w})]\} \end{aligned} \tag{7}$$

- The optimal parameters $\hat{\mathbf{w}}$ can be obtained by minimizing the cross-entropy error function, namely, $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} E(\mathbf{w})$
- By setting a threshold 50% (other thresholds can also be chosen), we have a decision rule:

$$\hat{y}_n = \begin{cases} 1, & \text{if } g(\mathbf{x}_n|\hat{\mathbf{w}}) > 0.5 \\ 0, & \text{if } g(\mathbf{x}_n|\hat{\mathbf{w}}) < 0.5 \end{cases} \tag{8}$$

Performance Evaluation for Classification I

- True positive (TP): true label is positive $y_n = 1$, and estimated label is correct $\hat{y}_n = 1$. This case is referred to as "hit".
- True negative (TN): true label is negative $y_n = 0$ (sometimes we use $y_n = -1$), and estimated label is correct $\hat{y}_n = 0$.
- False positive (FP): true label is negative $y_n = 0$, and estimated label is wrong (positive) $\hat{y}_n = 1$. This case is referred to as "type I error".
- False negative (FN): true label is positive $y_n = 1$, and estimated label is wrong (negative) $\hat{y}_n = 0$. This case is referred to as "type II error".
- Sensitivity, recall, hit rate, or true positive rate (TPR), indicating the accuracy when the true label is positive:

$$\text{Recall} = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{9}$$

Performance Evaluation for Classification II

- Specificity or true negative rate (TNR), indicating the accuracy when the true label is negative:

$$\text{Specificity} = \frac{TN}{N} = \frac{TN}{TN + FP} \tag{10}$$

- Precision, indicating the accuracy when the predicted label is positive:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$

- Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Confusion Matrix

Sources: [20][21][

		Predicted condition		
	Total population = P + N	**Positive (PP)**	**Negative (PN)**	Informedness, bookmaker informedness (BM) = TPR + TNR − 1
Actual condition	**Positive (P)**	**True positive (TP),** hit	**False negative (FN),** type II error, miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$
	Negative (N)	**False positive (FP),** type I error, false alarm, overestimation	**True negative (TN),** correct rejection	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$

ROC curve I

Definition

- A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.
- The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

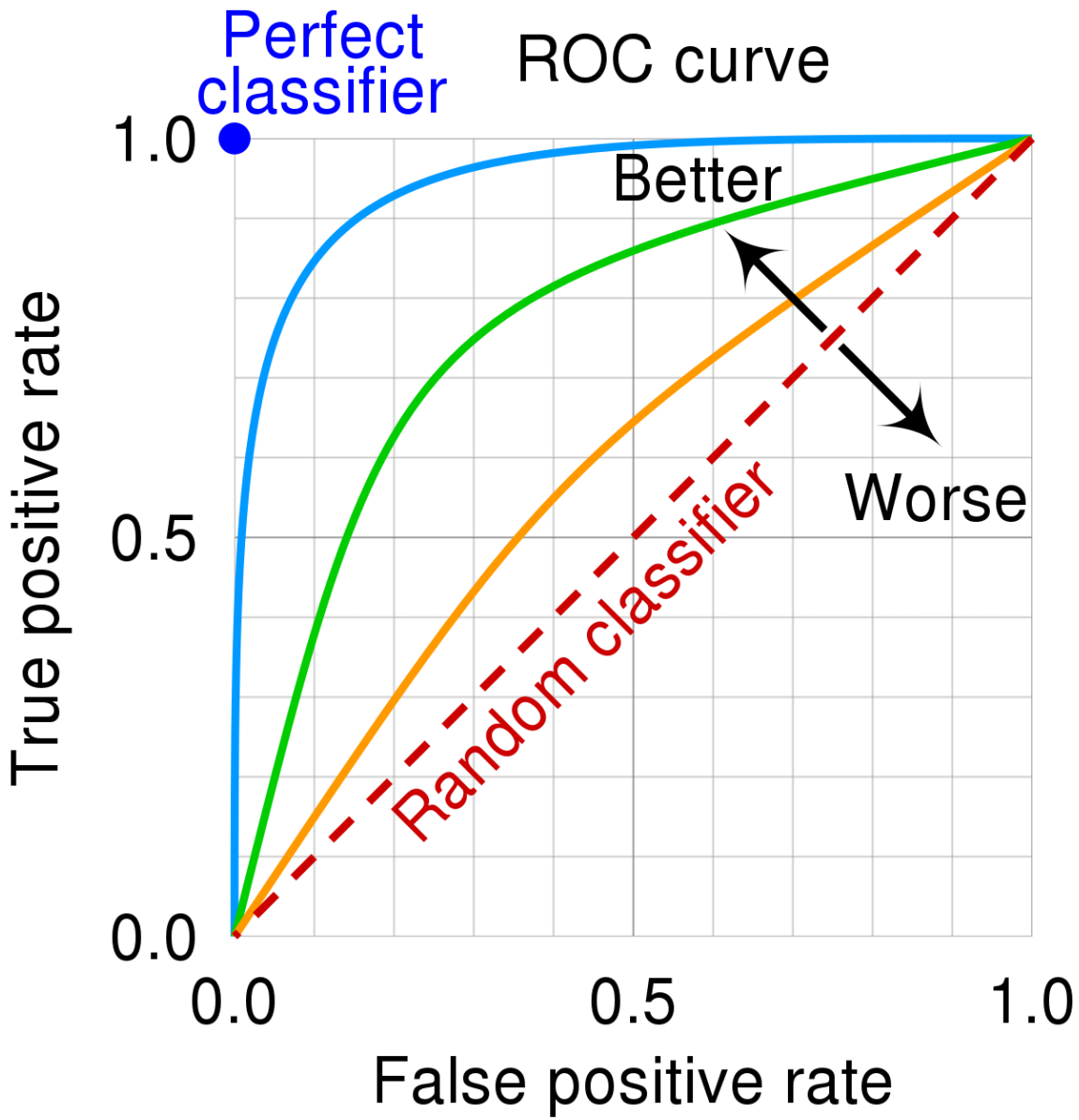
						TPR	FPR
True label	1	1	0	1	0		
Estimated probability	0.8	0.75	0.6	0.4	0.1		
Estimated label ($\tau = 0.5$)	1	1	1	0	0	$2/3$	$1/2$
Estimated label ($\tau = 0.7$)	1	1	0	0	0	$2/3$	0

ROC curve II

ROC space and curve

- By changing a sufficient number of thresholds and obtaining many data points of TPR and FPR, we then plot and connect the data points on a ROC space with TPR as the y axis and FPR as the x axis.
- The best performance corresponds to the point at which $TPR = 1$ and $FPR = 0$.
- The closer the ROC curve to the best point, the better the predictive performance of the classifier.
- AUC=the area under the ROC curve. The larger the AUC, the better the prediction performance.

ROC curve III



Dimensions of Supervised Learning

- Thee dimensions:
  - **Model**:

$$g(\mathbf{x}|\mathbf{w}) \tag{13}$$

  - **Loss function (Error function)**:

$$E(\mathbf{w}|\mathcal{X}) = \sum_{n=1}^{N} L(y_n, g(\mathbf{x}_n|\mathbf{w})) \tag{14}$$

  - **Optimization procedure/algorithm**:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} E(\mathbf{w}|\mathcal{X}) \tag{15}$$

- No free lunch theorem:
  - There is no universally best model.
  - Different types of models have to be developed to suit the nature of the data in real applications.

Unsupervised Learning

- No label is given, namely, only features $\mathbf{x}$ are available.
- Learning "what normally happens" and "interesting patterns" in the data.
- Much less well-defined than supervised learning with no obvious error measure.
- Some applications:
  - Density estimation
  - Clustering: grouping similar samples, e.g., K-means clustering, expectation-maximization algorithm.
  - Dimensionality reduction/data compression: to discover latent factors that can represent the majority of the information of the original data, e.g., principal component analysis.

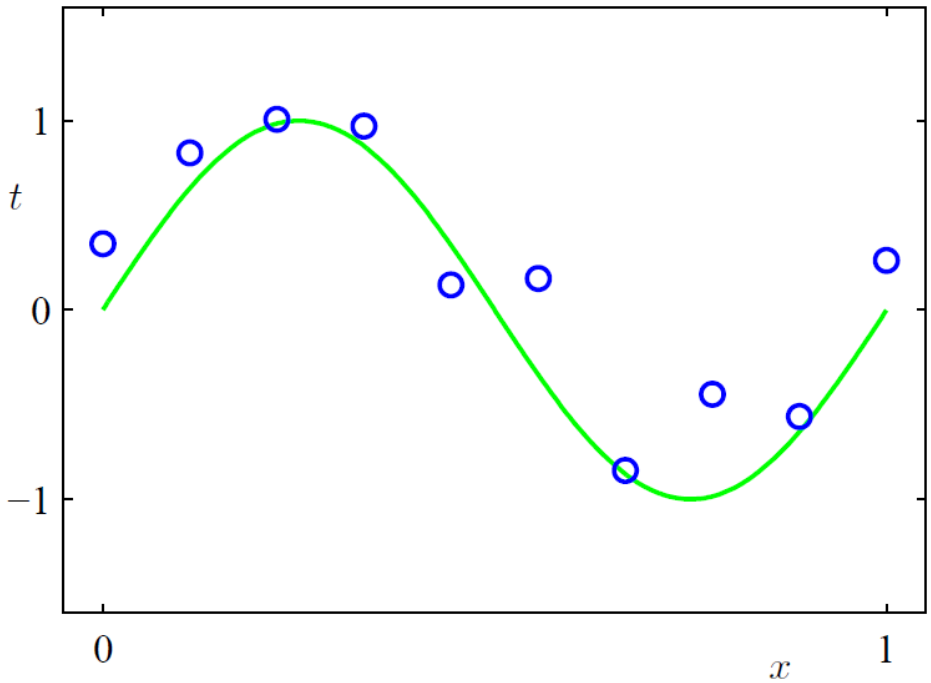
Reinforcement Learning

- Finding suitable actions to take in a given situation in order to maximize a reward in long run.
- There is a sequence of states and actions in which the learning algorithm is interacting with its environment.
- The current action not only affects the immediate reward but also has an impact on the reward at all subsequent time steps.
- The algorithm learn a policy (a sequence of actions) by a process of trail and error.
- Some applications:
  - Playing games, playing chess (e.g, Alpha Go)
  - Robot navigation in search of goal location
  - Traffic signal control
  - Order dispatching for ride-hailing services (Uber, Didi)

Case: Polynomial Curve Fitting

- Let us look at a regression problem:
  - Features: x (only one feature)
  - Label: t
- Data generation process (the same for training and test set):

$$t = \sin(2\pi x) + \text{noise} \tag{16}$$



Polynomial Function as Linear Model

- Then we use the following polynomial function to fit the training data:

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + ... + w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{17}$$

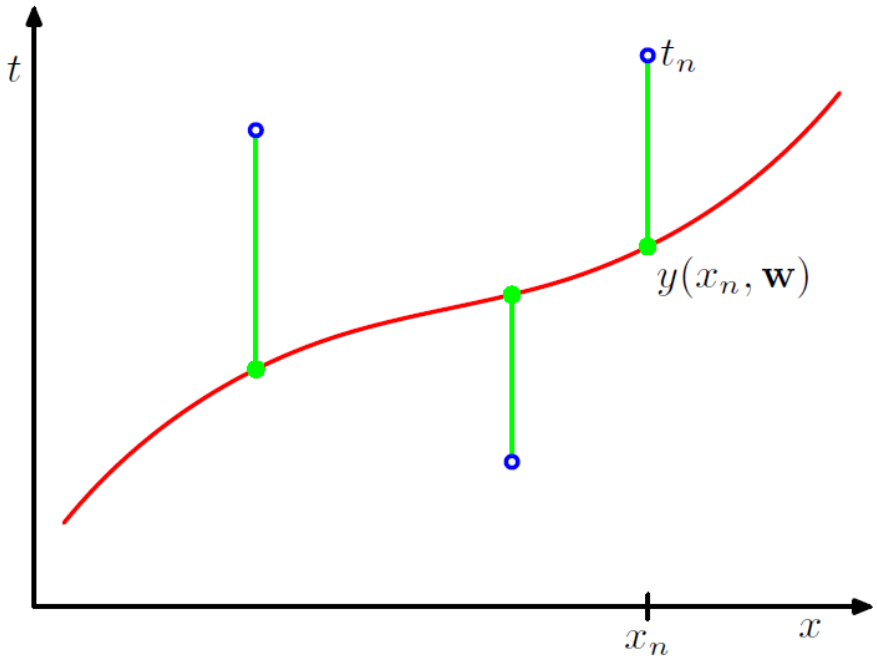
  where $\mathbf{w} = (w_0, w_1, w_2, ..., w_M)^T$ and M is the order of the polynomial.
- Linear model: the function $y(x, \mathbf{w})$ is nonlinear in x, but linear in $\mathbf{w}$.

Error Minimization

- Clearly, this is a regression problem, thus the error function is given by:

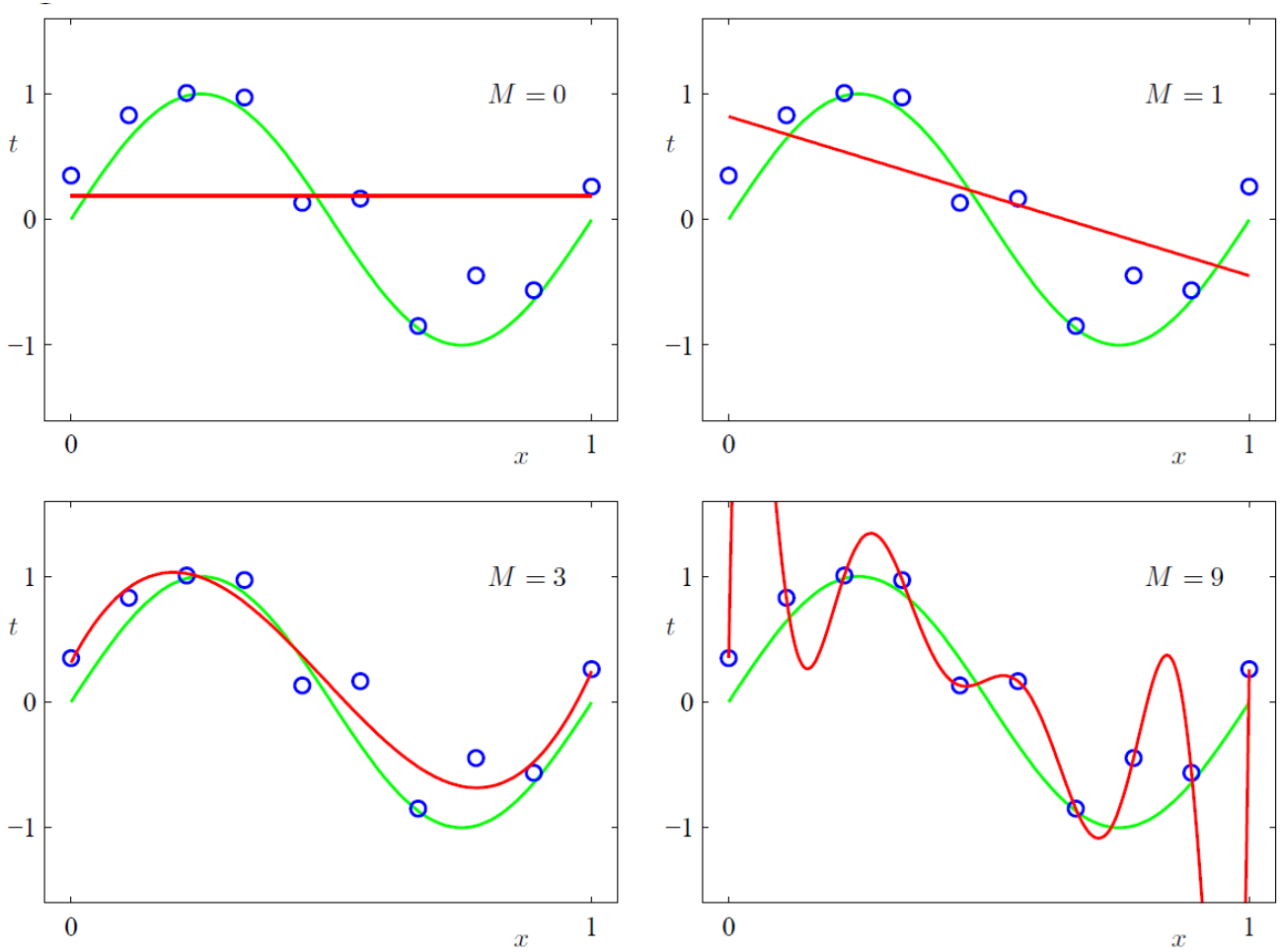
$$E(\mathbf{w}|\mathcal{X}) = \frac{1}{2}\sum_{n=1}^{N}[y(x_n, \mathbf{w}) - t_n]^2 \tag{18}$$

- The optimal parameters $\hat{\mathbf{w}}$ can be obtained by minimizing the error function $E(\mathbf{w}|\mathcal{X})$.

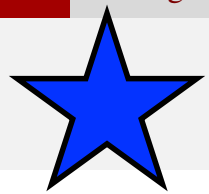


Model Selection

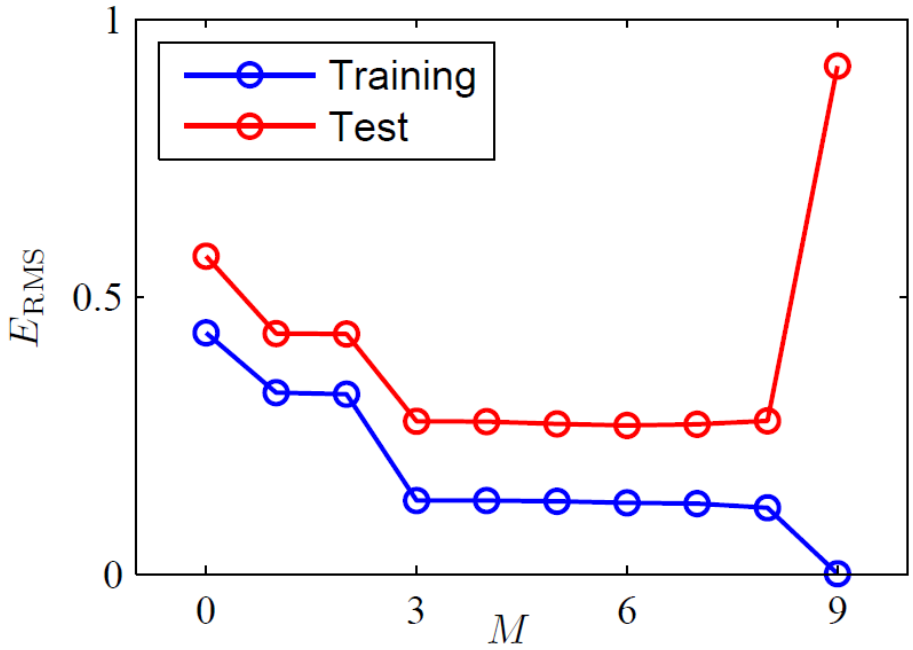
- Choosing the order of polynomial is an example of model selection. In this case, which model is better?



Over-fitting Issue 过拟合



- Although the training set and test set are drawn from the same distribution, they have different sample points.
- A very complex model may over-fit the curve of the training set, and thus may not well fit the test set. (indicating a low generalization ability)
- Calculate $E_{RMS} = \sqrt{2E(\hat{\mathbf{w}}|\mathcal{X})/N}$ for training and test set for different models (with different orders of polynomial)



Over-fitting Issue

- The optimal parameters $\mathbf{w}^*$ ($\hat{\mathbf{w}}$) in different models:

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

How to Reduce Over-fitting

- Method 1: increase the number of samples.

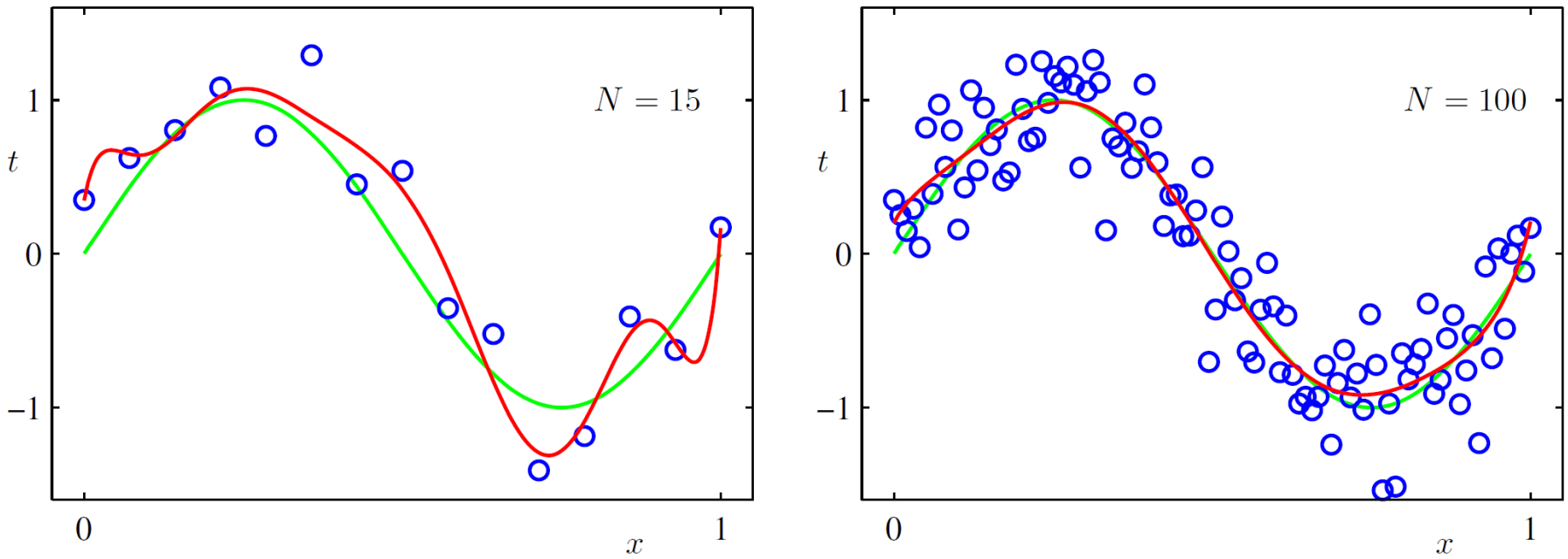


Figure 1.6 Plots of the solutions obtained by minimizing the sum-of-squares error function using the $M = 9$ polynomial for $N = 15$ data points (left plot) and $N = 100$ data points (right plot). We see that increasing the size of the data set reduces the over-fitting problem.

How to Reduce Over-fitting

- Method 2: Adding a regularization term to the error function:

$$\tilde{E}(\mathbf{w}|\mathcal{X}) = \frac{1}{2}\sum_{n=1}^{N}[y(x_n, \mathbf{w}) - t_n]^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 \tag{19}$$

- where $\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w} = w_0^2 + w_1^2 + ... + w_M^2$, and the coefficient λ governs the relative importance of the regularization term compared with the sum-of-squares error term.
- The larger the λ, the stronger the penalty on the size of parameters. This will force the model to select parameters with reasonable values.

How to Reduce Over-fitting

- No regularization: $\lambda = 0, \log \lambda = -\infty$
- Medium regularization coefficient: $\lambda = e^{-18}, \log \lambda = -18$
- Large regularization coefficient: $\lambda = 1, \log \lambda = 0$

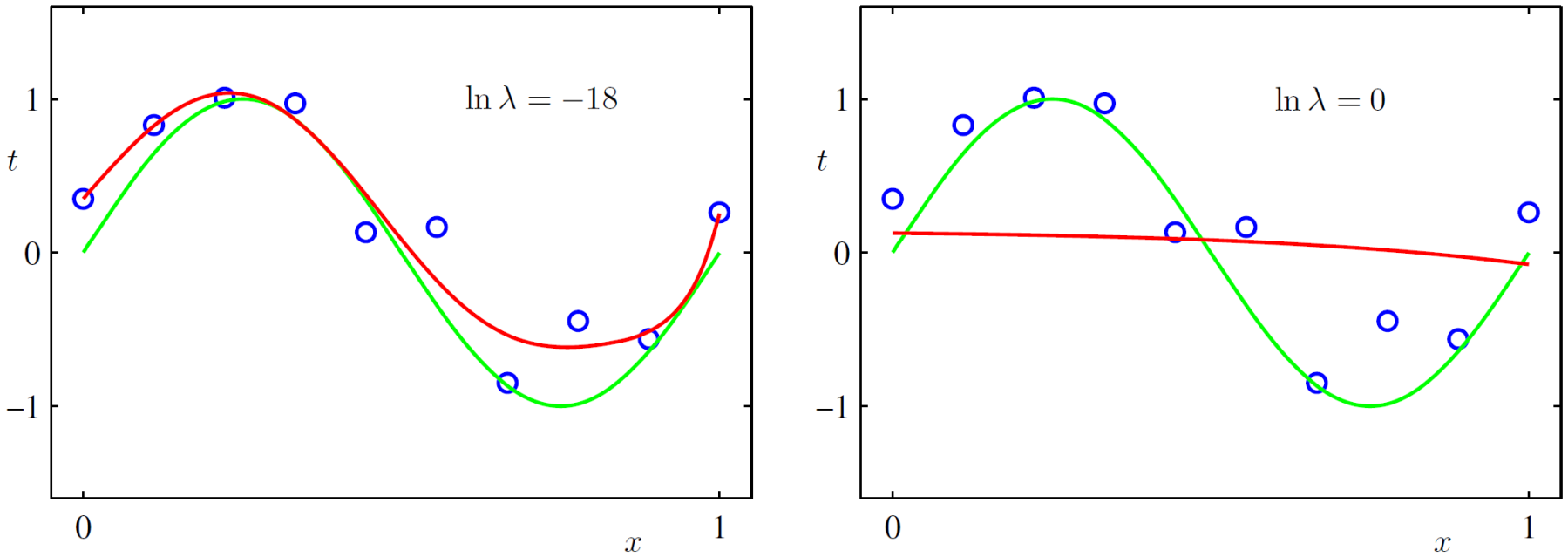


Figure 1.7 Plots of $M = 9$ polynomials fitted to the data set shown in Figure 1.2 using the regularized error function (1.4) for two values of the regularization parameter λ corresponding to $\ln \lambda = -18$ and $\ln \lambda = 0$. The case of no regularizer, i.e., $\lambda = 0$, corresponding to $\ln \lambda = -\infty$, is shown at the bottom right of Figure 1.4.

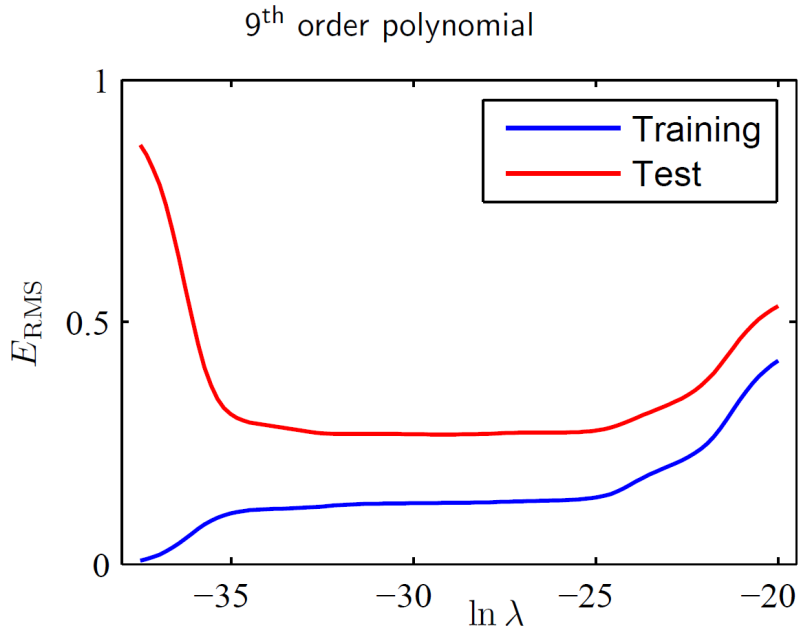
How to Reduce Over-fitting

- We see that, as the value of λ increases, the absolute magnitude of the coefficients gets smaller.

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

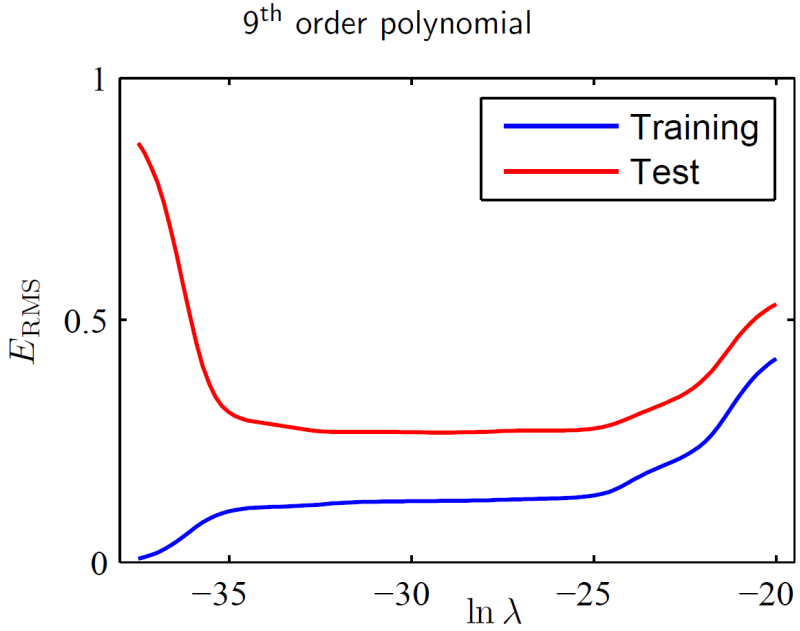
How to Reduce Over-fitting

- As the value of λ increases, the gap between the RMSE in the training and test set becomes more narrow.
- The generalization error (test error) first decreases and then increases with λ, indicating that there exists a trade-off between bias and variance.
  - Bias: how much the expected value of the label diverge from the true label
  - Variance: how much, on the average, the estimate varies from the expected value of the label



How to Reduce Over-fitting

- Under-fitting: when $\log \lambda$ is very large, the training error itself is large, leading to a large bias. Thus the test error is also large.
- Over-fitting: when $\log \lambda$ is very small, although the training error is very small, the model has very large variance, leading to a bad generalization ability or a large test error.
- Good fitting: when $\log \lambda$ is medium, the model has medium training error and low test error.



Cross-Validation

交叉验证！！！ 一般都采用这种方法，因为我们*SHM*的有效数据都是不足的～所以要尽量物尽其用～

- To estimate the generalization error, we need data unseen during model training.
- As usual, we split the data into:
  - Training set (e.g., 50%)
  - Validation set (e.g., 25%)
  - Test set (e.g., 25%)
- When data is limited, we may use k-fold cross-validation
  - Shuffle the dataset randomly.
  - Split the dataset into k groups
  - For each unique group, do:
    - take the group as a test set and the remaining groups as a training set,
    - fit the data with the training set and evaluate it on the test set,
    - record the test error.
  - Calculate the average test error of all groups.

Outline

Installation of Python

- Please refer to the instruction file (available on HKU Moodle)
- (*Recommend but not necessary*) Setup a designated virtual environment for CIVL7018. (See steps in the instruction file)

Useful packages

Jupyter Notebook

jupyter, ipdb

Visualization

matplotlib.pyplot, seaborn

Scientific Computing

numpy, scipy, pandas, scikit-learn (sklearn)

Deep Learning Frameworks

PyTorch (torch), tensorflow

Useful packages - Cont'd

How to install

```
pip install XXXX
```

How to use (in Python command line, or .py file, or Jupyter Notebook inline)

```
import XXXX
```

, where XXXX is the package name (see the previous slide, for instance, scikit-learn). [1]

[1] Note that when importing scikit-learn, you should type "import sklearn"

Outline

Kaggle

- Kaggle[2] offers a no-setup, customizable, programming environment and various datasets for you

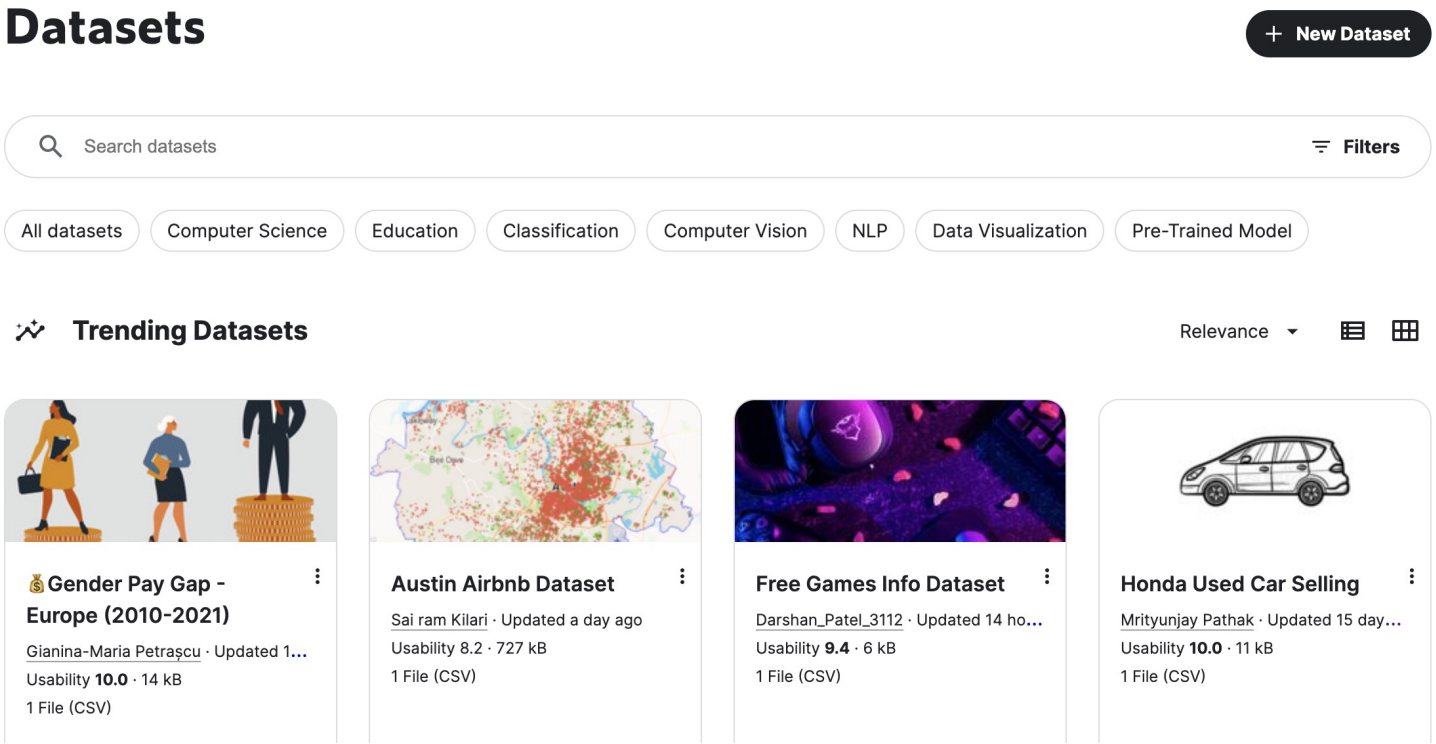


Figure: Kaggle

[2] In this slideshow, you can click the figure name to get access to the targeted website

Machine Learning Dataset from UC Irvine

- UCI machine learning repository is a well-known database for standard testing, where you can easily get data that fits your specific requirements

Figure: UCI Machine Learning Repository

Registry of Open Data on AWS

- Amazon Web Services (AWS) is the most comprehensive and widely used cloud platform in the world, and it also offers a public registry to help people discover and share datasets that are available via AWS resources

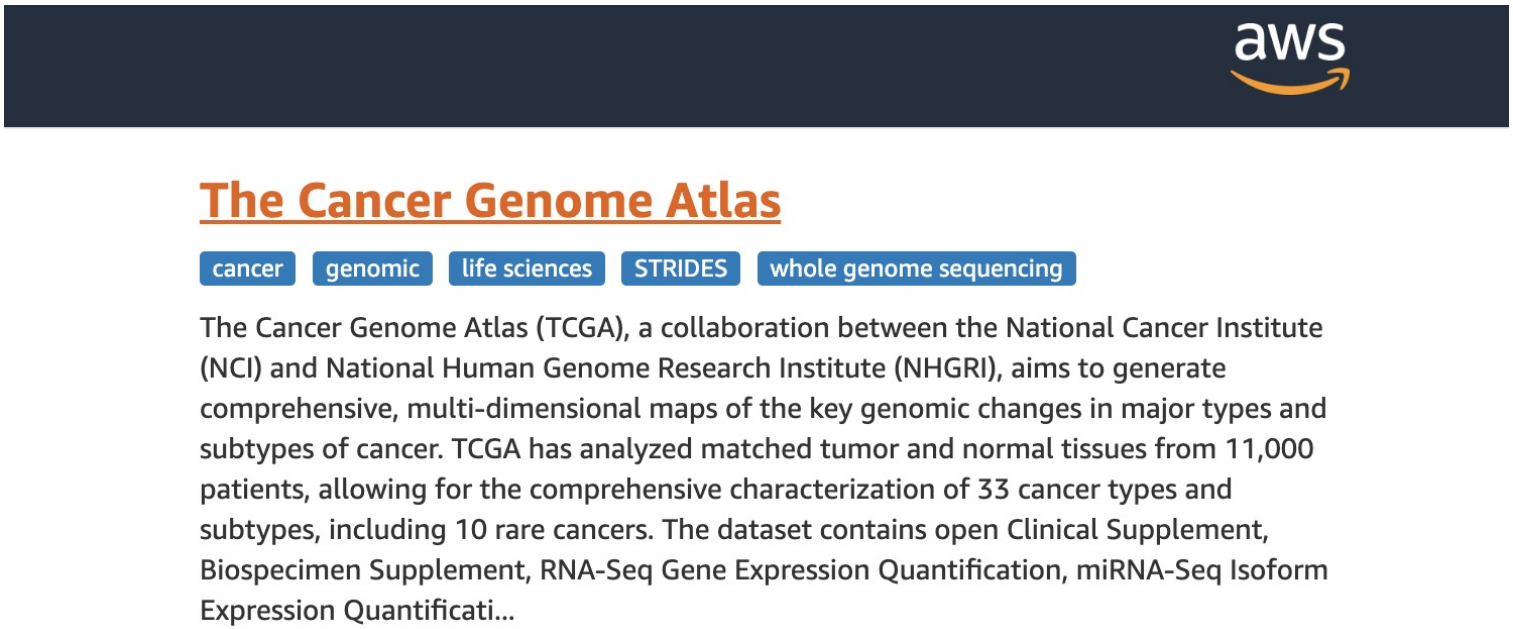


Figure: Registry of Open Data on AWS

Microsoft Research Open Data

- Microsoft research team provides a series of open datasets related to cutting-edge scientific research

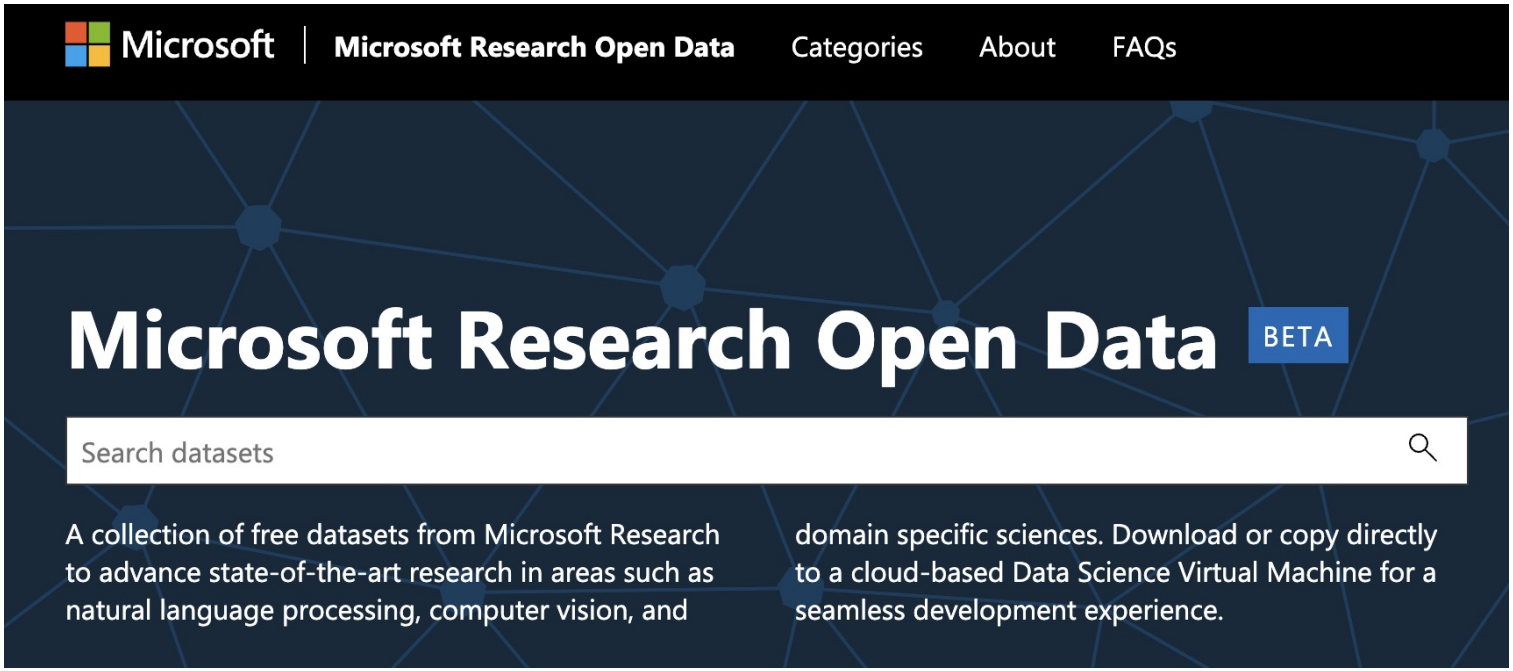


Figure: Microsoft Research Open Data

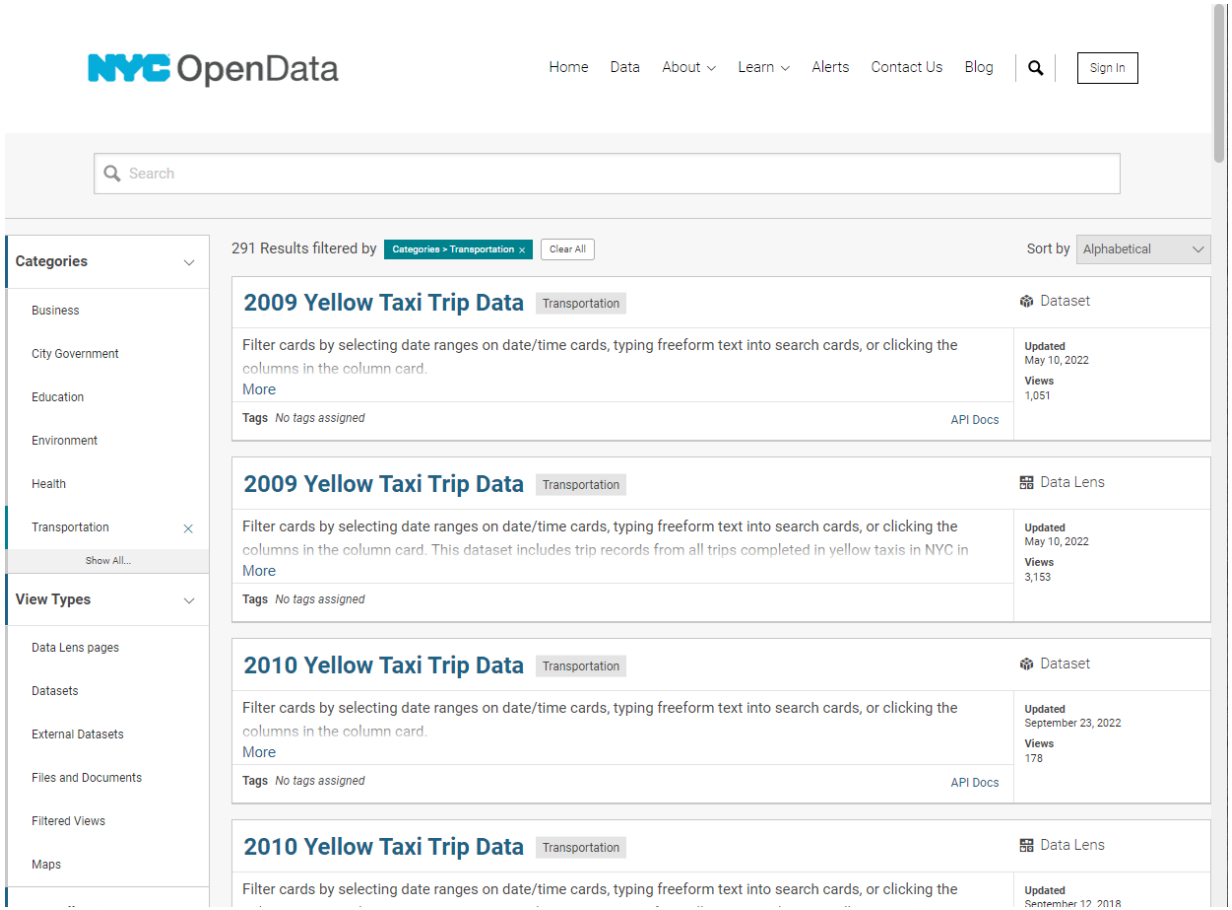
Open-sourced Data

Researchers can retreat data from a list of open-sourced data platforms, including but not limited to:

- NYC open data (https://opendata.cityofnewyork.us/)
- Hong Kong open data (https://data.gov.hk/sc/)
- Caltrans PeMs (https://pems.dot.ca.gov/)
- Didi GAIA Open Dataset
- RITIS (https://ritis.org/intro)
- TIMS (https://tims.berkeley.edu/)
- SafeGraph (https://www.safegraph.com/)
- Highways England Traffic Flow Data (https://webtris.highwaysengland.co.uk/)
- BART Ridership Data (https://www.bart.gov/about/reports/ridership)
- Bureau of Transportation Statistics (https://www.bts.gov/)

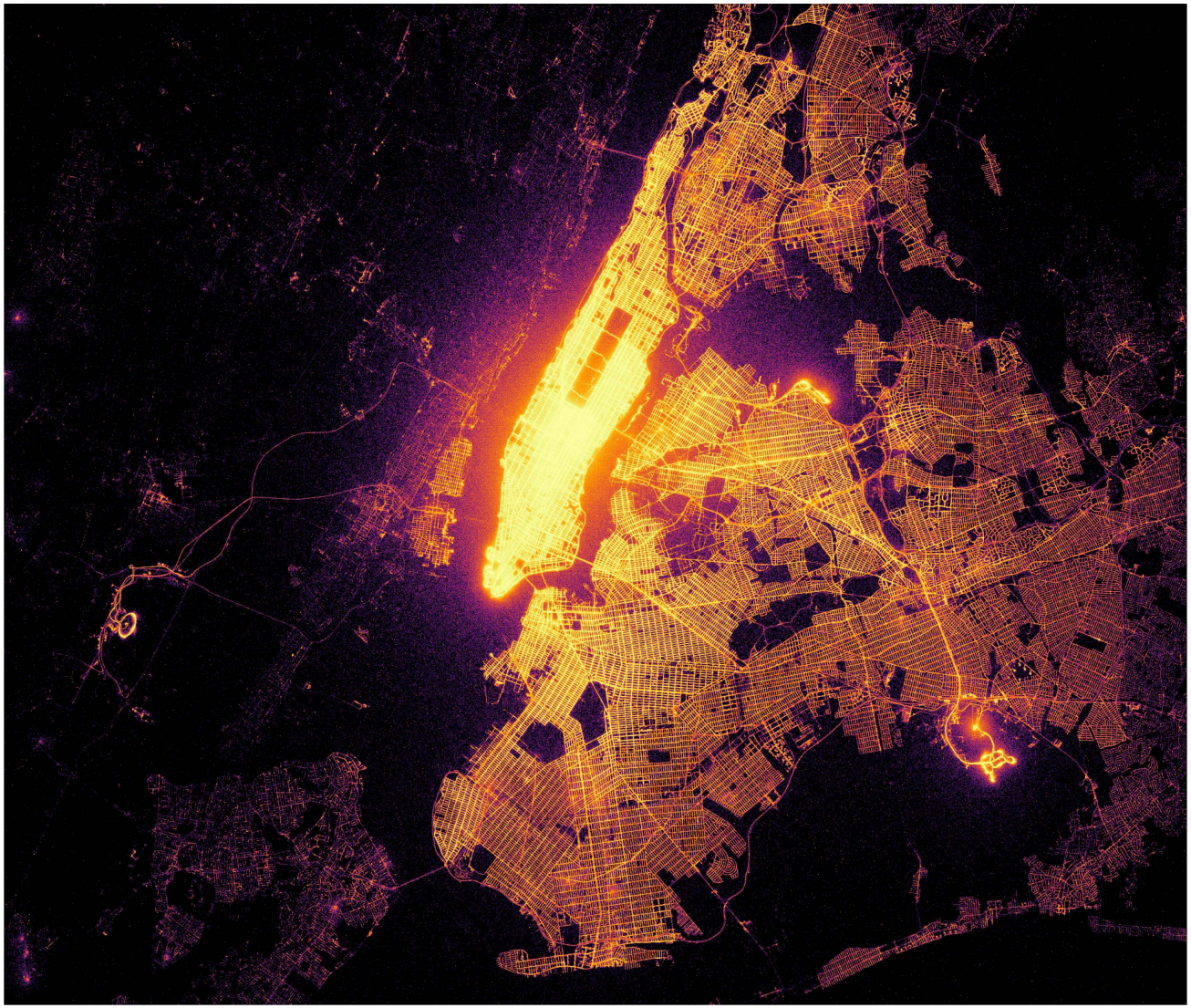
NYC Open Data

New York City open data platform (https://opendata.cityofnewyork.us/) offers many types of transportation data, including Real-Time Traffic Speed Data, Traffic Volume Counts, Bicycle Routes, etc., to the public.

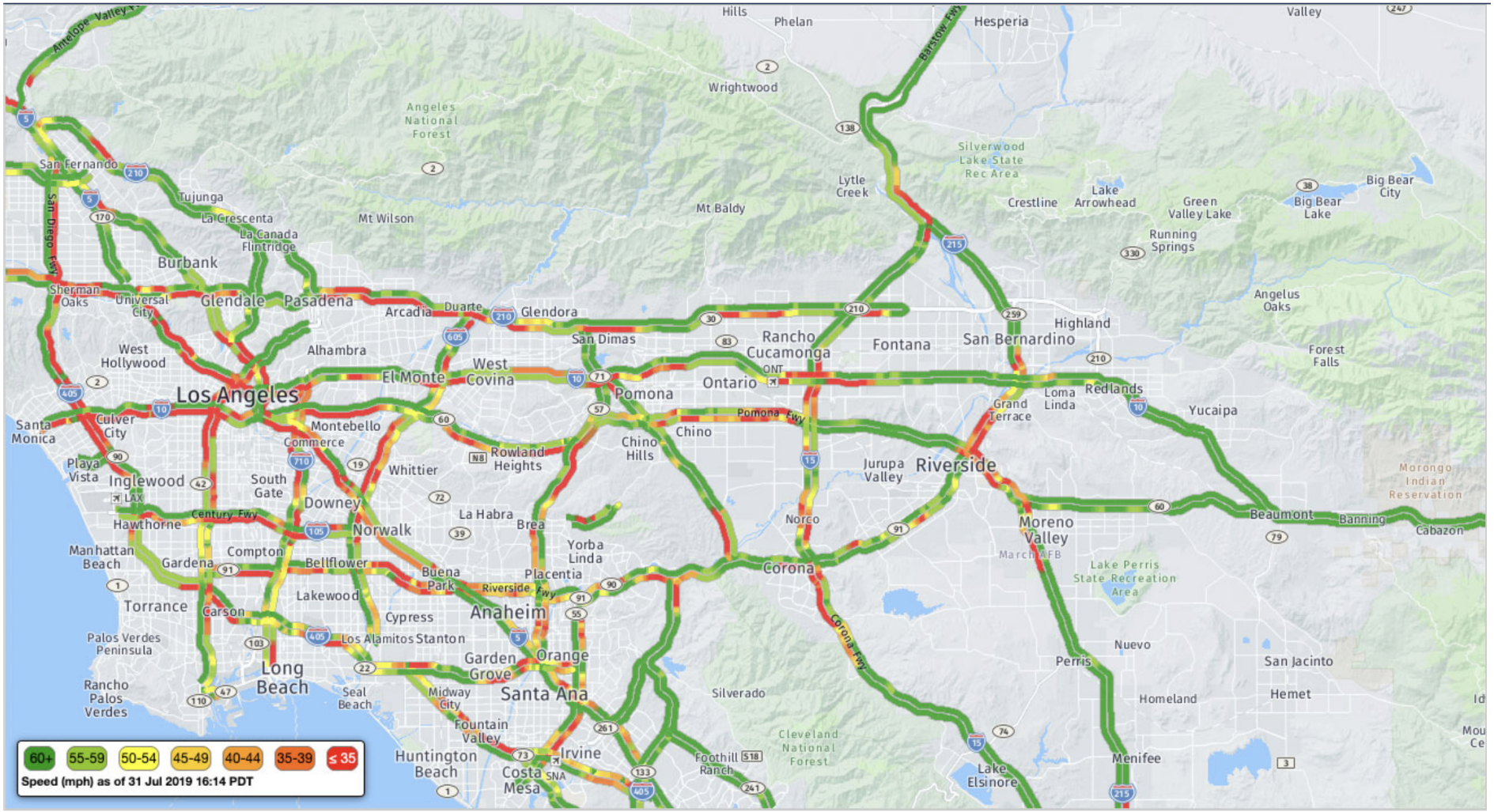


NYC Network

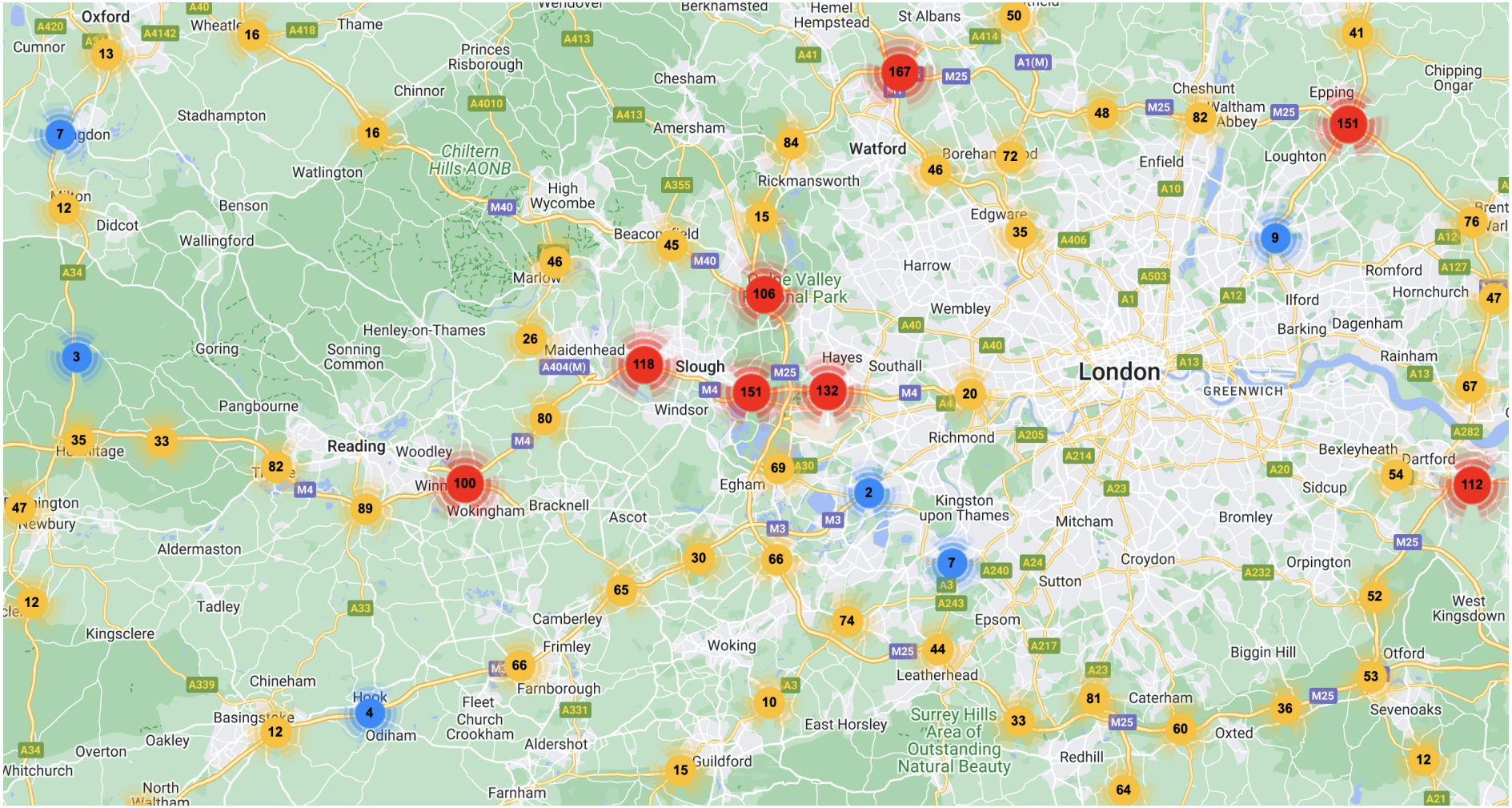
NYC Taxi Demand Intensity



Caltrans PeMS Data

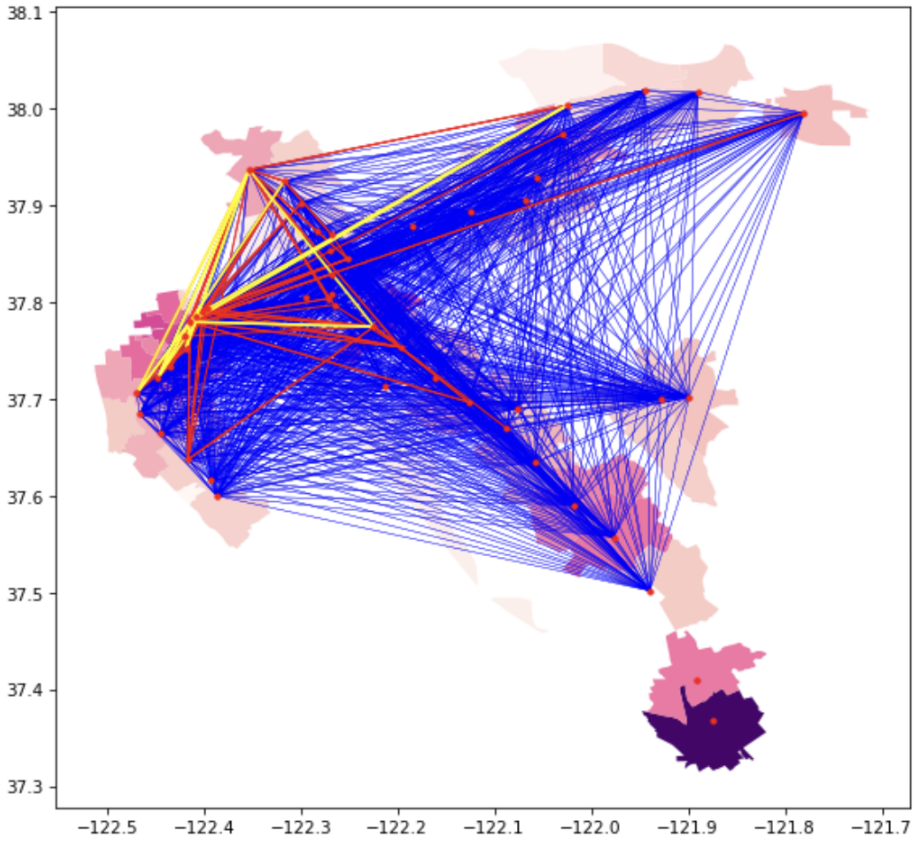


Highways England Traffic Flow Data



BART Ridership Data

The Bay Area Rapid Transit system provides ridership reports and data. Yellow lines represent high ridership (OD pairs), red lines represent medium ridership, and blue lines represent low ridership.



Other Useful Data I

The Weather Data Environment (WxDE)

- `https://wxde.fhwa.dot.gov/`
- The Weather Data Environment (WxDE) is a research project that collects and shares transportation-related weather data with a particular focus on weather data related to connected vehicle applications.

ITS Knowledge Resources

- `http://www.itsknowledgeresources.its.dot.gov/`
- ITS provide a proven set of strategies for advancing transportation safety, mobility, and environmental sustainability by integrating communication and information technology applications into the management and operation of the transportation system across all modes.

Other Useful Data II

Archived Data User Service (ADUS)

- `https://www.fhwa.dot.gov/policyinformation/travel/adus.cfm`
- ADUS provides the National ITS Architecture with the requirements for archiving and re-use of data collected for ITS operations. This FHWA website provides a wealth of information related to ITS and traffic related data.

Data.gov

- `https://www.data.gov/`
- The purpose of Data.gov is to increase public access to high value, machine readable datasets generated by the Executive Branch of the Federal Government.

Other Useful Data III

US DOT ITS JPO

- `http://www.its.dot.gov/`
- The US Department of Transportation (US DOT) ITS Joint Program Office (JPO) home page.

National Transit Database (NTD)

- `http://www.ntdprogram.gov/`
- NTD is the Federal Transit Administration's (FTA) national database of statistics for the transit industry. The NTD is composed of data reported by more than 600 transit agencies across the US. The data are then analyzed and compiled into reports published by FTA and made available to the public on the NTD Program website.

Other Useful Data IV

Traffic Volume Trends

- `http://www.fhwa.dot.gov/policyinformation/travel_monitoring/tvt.cfm`
- Traffic Volume Trends is a monthly report based on hourly traffic count data reported by the States. These data are collected at approximately 4,000 continuous traffic counting locations nationwide in US.

TranStats - Multimodal Transportation Database

- `http://www.transtats.bts.gov/`
- TranStats provides one-stop shopping for intermodal transportation data. It has a searchable index of more than 100 databases covering every mode of transportation, plus social and demographic data commonly used in transportation research.

Traffic Simulation Platforms I

Transportation Analysis and Simulation System (TRANSIMS)

- `http://code.google.com/p/transims/`
- TRANSIMS is an integrated set of tools developed to conduct regional transportation system analysis. An open source community has been developed as an independent and self-governing collaboration of TRANSIMS users, researchers and developers.

Simulation of Urban MObility (SUMO)

- `https://www.eclipse.org/sumo/`
- SUMO is a free and open source traffic simulation suite. It is available since 2001 and allows modelling of intermodal traffic systems - including road vehicles, public transport and pedestrians.

Traffic Simulation Platforms II

Flow: A Deep Reinforcement Learning Framework for Mixed Autonomy Traffic

- `https://github.com/flow-project/flow`
- Flow is a computational framework for deep RL and control experiments for traffic microsimulation. It is developed by a team in UC Berkeley led by Alexandre Bayen.

CityFlow

- `https://github.com/cityflow-project/CityFlow/`
- CityFlow is a multi-agent reinforcement learning environment for large-scale city traffic scenario. It can simulate the behavior of each vehicle, providing highest level detail of traffic evolution, and provide friendly python interface for reinforcement learning.