ML and Data

Supervised Learn map between input and labelled output data: predict output for unseen input.
Unsupervised Knowledge discovery or anomaly detection from input data.
Dataset is large and diverse. Suffers from *curse of dimensionality* due to large p and small n. Sparse feature spaces overfit. Increase in p necessitates complex models, which requires more data.
Training Training model.
Validation Tuning hyperparameters.
Test Evaluating model.
If data is limited, use *cross-validation* to dynamically split data (80–20) so that each split is used in validation, then choose best model.
Normalisation Scale between $[0, 1]$.
Standardisation Mean 0, variance 1.
Encoding Convert categorical data to numerical.
Mean-Centring Subtract mean from data.

Supervised Learning

Linear Regression

$x^{(i)}$-predictor, $y^{(i)}$-response. Strength of linear relationship between x and y using covariance s_{xy}:

- $s_{xy} > 0$: y increases as x increases
- $s_{xy} < 0$: y decreases as x increases
- $s_{xy} \approx 0$: no linear relationship

To generalise, use normalised correlation coefficient r_{xy} between -1 and 1. r has no information about gradient and $r = 0$ only implies no *linear* relationship.
Linearity Relationship is indeed linear. Scatter plot ($\hat{y}$ vs. y).
Multicollinearity $x^{(i)}$ are independent. $\left|r_{x_1,x_2}\right| > 0.7$ suggests high correlation.
Independence $y^{(i)}$ are independent.
Normality $\epsilon^{(i)} \sim \mathcal{N}\left(0, \sigma^2\right)$. QQ-plot and histogram of ϵ.
Homoscedasticity $\epsilon^{(i)}$ have constant variance. Residual plot (ϵ vs. $\hat{y}$).
Exogeneity No correlation between $x^{(i)}$ and $\epsilon^{(i)}$. Residual plot (ϵ vs. x).
R^2 Proportion of variance explained by predictors.
Adjusted R^2 Adjusts for #predictors.
F-statistic Overall significance of model (<0.05).
p-value Significance of β_i (<0.05). Misleading if assumptions are violated.
Regularisation prevents overfitting by regularising coefficients. **Bias** is error introduced by erroneous assumptions in model. High bias *underfits* training data. Increasing complexity reduces bias.
Variance is error introduced by model's sensitivity to fluctuations in data. High variance *overfits* training data. Reducing complexity reduces variance.
LASSO (L1) $\beta_i \to 0$: feature selection. Faster during testing.
Ridge (L2) Reduces large β_i, smoother cost function. Faster during training.

Classification

Predicting classes of categorical data.
SVM (Binary) separates classes with max margin. Use kernel when non-linear. Optimal classifier fails when not linearly separable and is sensitive to outliers. Slack variables allow misclassification. Small box constraint C allows more misclassifications.
KNN classifies points using majority class of k nearest neighbours. No training phase, sensitive to size of data and outliers. Small k leads to poor performance on noisy data. Large k leads to misclassification when points are close to decision boundary or classes imbalanced. Distance metric is problem dependent. Distance weighting can be applied to give more weight to closer points.
Random forest combines output of multiple decision trees using random subsets of training data. Data is split by class purity (Gini inpurity) or information gain. Tree depth should consider class imbalance; large tree depths often overfit. More trees provide better average predictions.
One-vs-One pairs class with another class ($p\left(p+1\right)/2$ classifiers).
One-vs-All pairs class with all other classes (p classifiers). Suffers from class imbalance—use weights inversely proportional to size of class.
Evaluation $\begin{bmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{bmatrix}$

- Recall: minimise false negatives.
- Precision: minimise false positives.
- Accuracy: overall performance.
- F1-score: harmonic mean of precision and recall.

Consider class imbalance as metrics may be biased toward majority class.
Select model using grid-search to tune hyperparameters.

Neural networks

Loss functions decide how errors are penalised during training:

- **Regression** MSE, MAE
- **Classification** Binary CE (Logistic Loss), Hinge Loss; Categorical CE (one-hot), Sparse Categorical CE (integer)

Output specified as probability or logits (after or before softmax).
Activation adds non-linearity.
Backpropagation updates weights to minimise loss using gradient descent optimisers. **Epochs** represent time taken for one gradient descent step on training set: updates per epoch $= n/b$.
Number of epochs consider if validation accuracy continues to increase, and if training accuracy is significantly higher than validation accuracy (overfitting).
Small batches take suboptimal paths toward minimum as batches are less representative of dataset; may lead to underfitting when classes are imbalanced.
Large batches may require more epochs for satisfactory result and consume more memory.
Ensemble methods leverage multiple models to improve performance but increase computational cost.
Regularisation performed to prevent overfitting by using dropout layers, batch normalisation (layers trained on same scale, improves training speed), weight regularisation (L1/2 on weights, biases, activation functions), or fine tuning existing models. **Augmentation** increases size of training set. Should not change meaning of image, nor be too extreme that it is unlikely.

Convolutional Neural Networks

Learn spatial features using kernels. Stack 1/2 convolution, max pooling and batch normalisation layers with increasing filter size, followed by fully connected layers. **Residual networks** introduce skip connections from input layer to output of convolution layers (fast training when deep). When number of layers (parameters) is large, use bottleneck layers to keep internal representation small (1×1 filter at start/end of convolution block, or in skip branch: patterns across channels). Addresses vanishing gradient problem.
Metric Learning learns embeddings for verification and identification, where similar classes are close, and dissimilar classes are far. Can add new classes after training. Backbone network is an encoder that is trained to learn embeddings.

- Siamese passes pairs of images through network. Uses binary CE loss (does not force similar images to be close) or contrastive loss.
- Triplet passes triplets (anchor, positive, negative) of images through network. Uses triplet loss (similar to contrastive loss).

Contrastive and triplet loss separate pairs by a margin. Normalise embedding to use a margin of 1. Distribution of distances should minimise overlap.
Embedding size should be sufficiently large and depends on number of classes and complexity of backbone.

Unsupervised Learning

K-Means clustering finds k clusters that minimise distances between data points and their cluster centres without overlap. Run multiple times as clusters are sensitive to initialisation of cluster centres (pronounced when number of data points/clusters increases). Distance metric is problem dependent but restricts algorithm to roughly spherical clusters. Scale dimensions to avoid domination.
GMMs assign a probability that data belongs to a cluster. Initialise with k-means. Computationally expensive.
k-means is only used for exploration, as it doesn't measure cluster likelihood. GMMs can be used for exploration and detecting abnormalities. k-means is more efficient and scales better than GMMs.
Determine k using reconstruction error (average distance to assigned cluster; elbow; only for k-means) or BIC (model informativeness; minimum).
Under-clustering true clusters merge into super-cluster.
Over-clustering true cluster divided into sub-clusters.
Dimensionality reduction: good for sparse data; reduces computational cost by removing less important features.
PCA projects data onto orthogonal components that maximise variance. Used to extract features and compress data. Choose the top q features using cumulative sum of explained variance (components are sorted by importance). $q = p$ retains all information. Must standardise data before applying PCA and ensure $n \gg p$.
LDA (supervised) finds projection that best discriminates between two classes. Returns $k-1$ components. Sensitive to number of samples per class: each class should have more samples than number of features. Large n or p leads to poor performance, due to overfitting. First apply PCA. Unlike PCA, LDA cannot reconstruct original data.
t-SNE (supervised) visualise data in 2D using locally linear relationships.
Autoencoders reconstruct input data (regression/MSE). The encoder compresses inputs to a lower-dimensional bottleneck using convolution and max pooling with decreasing filter size.
The decoder then reconstructs input using convolution and upsampling with increasing filter size. Consider dimension of input for bottleneck.
Multi-Task learning uses common encoder to train multiple networks. Weight loss functions for each task to prevent task from dominating. Each task regularises the others. Ensure correct loss is used for each task.
Semi-Supervised learning uses both labelled and unlabelled data in training. Loss function does not include unlabelled data.
Variational autoencoders generate data by learning a continuous latent space which we can sample from using a decoder. Use KL divergence to constrain distribution to be normal (samples distributed across latent space).

Sequences and Attention

Sequence data is time-dependent and has variable length.
RNNs process sequential data using feedback loops, where previous time-step information is passed to the network with the next time-step. Suffers from vanishing/exploding gradient problem.
LSTM uses sigmoid and tanh functions to control flow of information. Long-term memories are updated using forget gate. Short-term memories are updated using input and output gates. Placed after embedding layer. May be stacked to improve performance, but can be expensive; first LSTM layer should return the sequence to the next LSTM but the last LSTM should return a vector of dimension appropriate for sequence.

- Sequence-to-one: sequence input and single output: $X_{t+1} \mid X_t, \dots, X_0$.
- Sequence-to-sequence: sequence input and sequence output: $X_{t+T}, \dots, X_{t+1} \mid X_t, \dots, X_0$. Can be used for regression or classification.

Bi-directional LSTMs process data forward and backwards through the network to learn from past and future.
Attention mechanism allows network to focus on important elements of a sequence. Attention weights are learned during training. Placed between encoder and decoder. Activation function can be softmax or sigmoid, depending on how individual parts of the sequence should be weighted. Attention weights can be visualised to understand what the network is focusing on.
Transformers use multi-headed self-attention to measure the importance of each element in a sequence in relation to all other elements. Multi-headed refers to learning multiple ways to learn attention. Outperform LSTMs, but require far more operatios. Can be stacked. Outputs are always sequence-to-sequence, but can use dense layers or global average pooling to reduce sequence to vector.
Transformers can be parallelised unlike RNNs and LSTMs which are sequential.