What is data science? Data science is a field that extracts insight from data. This involves collecting/preprocessing data, analyzing data, and using data to build predictive models (this is the machine learning part, although data scientists typically work on offline models). Ultimately, the goal is to generate insights that solve a business need. Domain knowledge.

What’s involved in data analysis?

What’s involved in data engineering?

What’s involved in machine learning engineering? This can involve both offline and online models. Typically, data scientists do not work on online models, deployment, monitoring, retraining, etc.

ML algorithm categories:

* Supervised, unsupervised, semi-supervised
  + Classification vs. regression vs. clustering
* Parametric vs. non-parametric algorithms
* Linear vs. nonlinear algorithms

Supervised learning:

* Linear algorithms
  + Linear regression
    - Least squares, residuals, linear vs. multivariate regression
  + Logistic regression
    - Cost function, sigmoid function, cross entropy
  + SVM
  + LDA (classification)

***Linear regression:***

* Hypothesis:
* Cost function:
* Gradient descent:
* If you assume that the errors, , are all Gaussian with zero mean (or equivalently, ), then
  + To maximize (or equivalently, minimize NLL), you need to minimize the least squares. LSE and MLE are equivalent.
* Normal equation:

***Logistic regression:***

* Hypothesis:
* Maximize /minimize NLL (aka binary cross entropy)
* Gradient descent:
* Sigmoid/hypothesis comes from exponential family/GLM formulation

***Perceptron:***

* Hypothesis:
* Step function; sigmoid is softer version of this
* vector is normal to the decision boundary and points in the direction of the positive samples
* Error is 0 or . During gradient descent, the vector is adjusted to be positively correlated with positive samples and negatively correlated with negative samples.

***Exponential families and generalized linear models:***

* Exponential family distribution has the form
  + is the base measure, a scalar
  + is the natural parameter, and is the sufficient statistic. They have the same dimensions, and for many distributions, .
  + is the partition function; it normalizes the distribution
* Properties of exponential family distributions
  + MLE of w.r.t. is concave (only one minimum, the global minimum). Equivalently, NLL is convex.
* GLMs follow from these assumptions and design choices
  + Assume that follows an exponential family distribution parameterized by
  + Choose : the natural parameter is linear w.r.t. the input features
  + During training, , also known as the log-posterior probability. The gradient descent update for all GLMs is .
  + During inference, the model prediction, , is equal to , the mean of the distribution of conditioned on . is the canonical response function, and .
* Bernoulli distribution for binary data; becomes logistic regression
  + Canonical form ( is canonical parameter):
* Gaussian distribution for real data; becomes linear regression
  + Assuming , canonical form:
* Softmax regression (aka cross-entropy minimization) is a generalization of logistic regression to more than 2 classes
  + Raw logits:
  + (Categorical) cross entropy loss function:

***LDA:***

* <https://scikit-learn.org/stable/modules/lda_qda.html>
* Generative learning algorithms: Instead of learning , learn . Then . Learn the feature distribution for each class/fit distributions to input features for each class.
* Does not include the offset term,
* Assume , the conditional distribution, is a multivariate Gaussian. For LDA, assume that have the same for all . This gives a linear decision boundary.
* For -dimensional data,
  + Instead of conditional likelihood in discriminative learning algorithms
* Unlike GLM, there’s no need for gradient descent/hyperparameter tuning. The parameters that maximize likelihood can be calculated in closed form from the training data (sample means and covariance).
* Inference:
* Logistic regression vs. LDA
  + is a sigmoid function in both, but the decision boundary is different because they are parameterized differently.
  + LDA implies sigmoid function, but the reverse is not true. LDA assumptions are stronger than logistic regression.
* Technically, you can use other distributions (doesn’t need to be Gaussian)
* How does LDA reduce dimensions? How does this differ from PCA?

***Naïve Bayes:***

* Assume the features are conditionally independent given class, i.e.
* The choice of distribution, , affects the model ( can be continuous or discrete-valued)
* If you choose , then this becomes equivalent to QDA where input features are conditionally independent

***SVMs:***

***Decision trees:***

* <https://scikit-learn.org/stable/modules/tree.html>
* Non-parametric supervised learning method for classification and regression
* Advantages
  + Interpretability (white box model)
  + Requires little data preparation (e.g. no need for data normalization)
  + inference time where is the number of training samples
  + Can validate model using statistical tests, which means you can account for the reliability of the model
* Disadvantages
  + Prone to overfitting: without any limitations, trees can be extremely large and complicated. Mitigation: pruning, min leaf size, max tree depth, ensembling.
  + Are piecewise constant approximations, which means they’re not good at extrapolation
  + Learning an optimal tree is NP-complete, so practical decision-tree learning algorithms are based on heuristic algorithms like greedy search where locally optimal decisions are made at each node. These cannot guarantee to return the globally optimal tree. Mitigation: training multiple trees in ensemble learner, where features and samples are randomly sampled with replacement (bagging).
  + Trees can be biased due to imbalanced classes – balance training data before learning tree
* How does pruning work? Is it better to use pruning or other regularization techniques? Use both?
  + <https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html#sphx-glr-auto-examples-tree-plot-cost-complexity-pruning-py>
* Sklearn – binary decision trees? How to split features into two subsets? Does not support categorical?
* ID3

Bagging

* Out-of-bag error

Boosting

* Decision stump

Why is it called Bayes?

Softmax, multilabel

Why is softmax GLM?

SVM – are there other kernel functions? Sort of automates feature selection. Like DPD? Not gradient descent? How are the support vectors chosen?

Represented theorem – the parameters of the model are linear transformations of data points. Quantified by . Support vectors have nonzero .

How does SVM do regression?

Understand convex optimization, quadratic optimization/programming, etc. terms even if you don’t understand the math. Is convex same thing as quadratic cost function? NN backprop is not convex.

Features should be on the same scale – feature normalization. Helps gradient descent run more quickly. Makes more sense for regularization.

Ng says you should never make any model decisions based on test data, including online

What the heck are statistical tests?

* <https://www.youtube.com/watch?v=qBigTkBLU6g&list=PLblh5JKOoLUK0FLuzwntyYI10UQFUhsY9>
* <https://en.wikipedia.org/wiki/List_of_statistical_tests#:~:text=Statistical%20tests%20are%20used%20to,so%20statistical%20tests%20in%20use>

Curse of dimensionality, number of features vs. number of samples. In the regularization lecture, Ng says he prefers number of samples >= 10\*number of features. Why does high number of features lead to overfitting?

Nominal, ordinal, one-hot

Here’s a dumb question: on the iris dataset, should I be able to get 100% accuracy on the test set (after splitting)?

End-to-end ML lifecycle

* Preprocessing
* EDA
* Missing values – data imputation
* Data augmentation

1. Always normalize your data
2. Model.train() and model.eval(), torch.no\_grad()
3. Batch data loading
4. When to use ROC/AUC vs. F1
5. Regularization
6. Initialization/random seed
7. Hyperparameter tuning

Bias-variance analysis? Model has two sources of information: assumptions (inductive bias) and data (variance). If dataset is small, then you need to increase bias of your model, otherwise you risk overfitting. Number of features matters too, but why? Curse of dimensionality?

<https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff>

When I ensembled ResNet + ViT, AUC didn’t improve. What would be the reason for this? Does that mean variance cannot be further reduced? Somehow the combination of the two models performed worse than each individually. Were they trained on the same training data? Should I have weighted the prediction by their confidence/probability?

What contributes to bias? Inductive bias? Regularization adds bias, can this be seen as inductive bias, e.g. in decision trees?

Does pretraining reduce bias?

Explain white box (decision tree) vs. black box (ANN)

Statistical tests

A/B test

Dimensionality reduction – PCA, ICA, feature selection

What is the use case for feedforward neural networks? How could you even begin to build a custom network? <https://www.reddit.com/r/MachineLearning/comments/lzoqjg/d_why_neural_networks_for_tabular_data_are_bad/>

Types of machine learning

* Supervised learning
  + Linear algorithms
    - Linear regression
    - Logistic regression
    - SVM
    - LDA – is the decision boundary always linear?
* Decision trees
* Ensemble methods
* Unsupervised learning → clustering
* Semi-supervised learning
* Parametric vs. non-parametric
* Linear vs. nonlinear

Modeling

* Inductive bias (aka learning bias): this is the type of model you choose and the model hyperparameters. This is your assumptions about how the input and output are related, e.g. if you choose a linear regression model, you are assuming that the output is a linear function of the input. Example: LDA vs. logistic regression (see CS229 notes).
* Overfitting/underfitting, variance/bias

Cost functions

* MSE
* R2
* KL divergence, cross entropy, entropy, gini loss

Regularization

* Label smoothing
* L1/L2, L1/L2 for neural networks

NLP

* Standard ways to encode text (one-hot, etc.)
* Stopwords
* NLTK
* Spacy

Feature importance, feature selection

Decoding

Greedy search

Beam search