Introduction to Machine Learning

EPI 7913

Khaled El Emam & Doug Manuel

with William Klement and Juan Li

Course Parameters - I

- Office hours 11:30am to 1:30pm on class-Tuesdays
- More details are available on-line with additional resources. This will be updated as well so keep an eye on it
- Assignments: 10% / 30% / 30% / 30%
- We are using R
- You will need an account on an on-line analytics platform (credentials are supplied by email): https://epi7913.ehealthinformation.ca/
- You can use R Studio if you want but we will have the examples, templates, and assignments only in Jupyter Notebook

Course Parameters - II

- This is an in-person course; there is going to be a virtual component but that is discretionary
- Students who are not auditing are expected to attend the in-person classes
- Except in programs and courses for which language is a requirement, all students have the right to produce their written work and to answer examination questions in the official language of their choice, regardless of the course's language of instruction
- The university regulation on academic fraud: https://bit.ly/3cOp0cU

Scope of Course

- The course is intended to be applied, and to cover practical techniques that will be useful in realistic settings; it is not a survey of methods
- The focus will be on:
  - Structured data (as opposed to text, images, voice, etc.)
  - Phenotypic data (e.g., clinical, administrative, surveys)
  - Mostly supervised learning
  - Diagnostic / Prognostic methods (prediction)
- This is a specific slice through ML, but will give you a useful set of tools and examples to start from
- Mostly focused on cross-sectional (tabular) data as opposed to longitudinal data

Course Outline - I

Week	Topics Covered
Week 1: 12 Sept	Introduction to machine learning 1.Supervised vs unsupervised 2.Loss functions 3.Gradient descent
Week 2: 19 Sept	Data exploration 1.Univariate visualizations 2.Bivariate visualizations 3.Tabulations
Week 3: 26 Sept	Basic models 1.train/validate/test 2.k-nn models 3.CART models 4.Hyper-parameter tuning
Week 4: 3 Oct	Model evaluation 1.Hyper-parameter tuning (continued) 2.Cross validation 3.Bootstrapping 4.Classification measures
Week 5: 10 Oct	Review and exercises

Course Outline - II

Week	Topics Covered
Week 6: 17 Oct	Data preparation I 1.Dealing with missingness 2.Imbalanced data 3.Calibration
Week 7: 31 Oct	Data preparation II 1.Coding categorical variables 2.Embedding layers
Week 8: 7 Nov	Advanced modeling I 1.Bagging and boosting 2.Model ensembles
Week 9: 14 Nov	Advanced modeling II 1.Model ensembles (contd.) 2.Multilayer Perceptron
Week 10: 21 Nov	Model explainability 1.Variable importance 2.Partial dependence plots 3.SHAP values
Week 11: 28 Nov	Deploying and publishing models 1.Reporting guidelines 2.Software as a medical device 3.Monitoring performance
Week 12: 5 Dec	Review and exercises

Assignments

Assigned	Expected to be Handed in by	Points	Description
26th September 2023	9th October 2023	10%	Students will get a dataset that they will be expected to perform a descriptive analysis on, interpret the results, and answer some questions about the dataset.
10th October 2023	6th November 2023	30%	The students will build a classification model and evaluate its prediction accuracy using different approaches and compare the accuracy results.
7th November 2023	27th November 2023	30%	The students will evaluate the performance of prognostic models using multiple methods and identify the most important variables.
28th November 2023	22nd December 2023	30%	The students will train and calibrate a prognostic binary classification model and identify the most influential variables.

JUPYTER NOTEBOOK

JupyterLab: Notebook Interface

- JupyterLab is a web-based interactive development environment for notebooks, code and data.
- Enables the manipulation of workflow in data science and machine learning
- It supports multiple programming environments including R programming.
- More on: https://jupyter.org/

What is the Jupyter Notebook?

- It is an interactive computing environment to enable users to produce notebook documents.
- Notebooks may contain live code, plots, narrative text, equations, images, videos, etc.
- Components:
  1. Notebook web application is used for writing and running code interactively
  2. Kernels separate processes for each notebook so they can run without getting mixed up
  3. Notebook documents are self-contained documents with associate code and kernel

Notebook documents

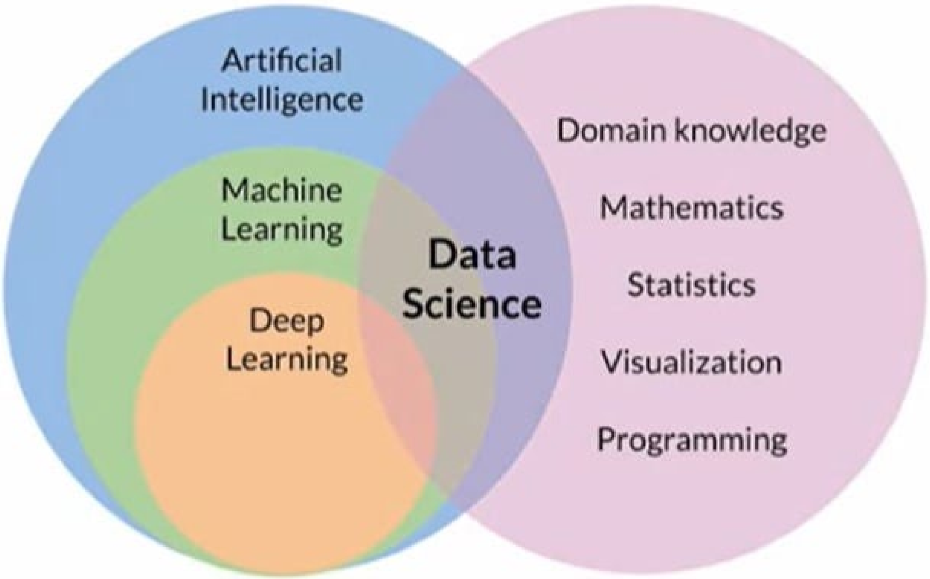
jupyter
fibonacci
data_plotting

- These are just files with .ipynb extensions
- Consist of a linear sequence of:
  - **Code cells** for live code that will be run in the kernel (in our case, the R kernel)
  - **Markdown cells** contain rich formatted text with embedded LaTeX equations
  - **Raw cells** for unformatted, unmodified texts
- Can be uploaded or downloaded
- Can be converted to other formats (HTML, PDF, etc.)

https://jupyter-notebook.readthedocs.io/en/latest/examples/Notebook/examples_index.html

INTRODUCTION TO MACHINE LEARNING

Part I: What is machine learning?



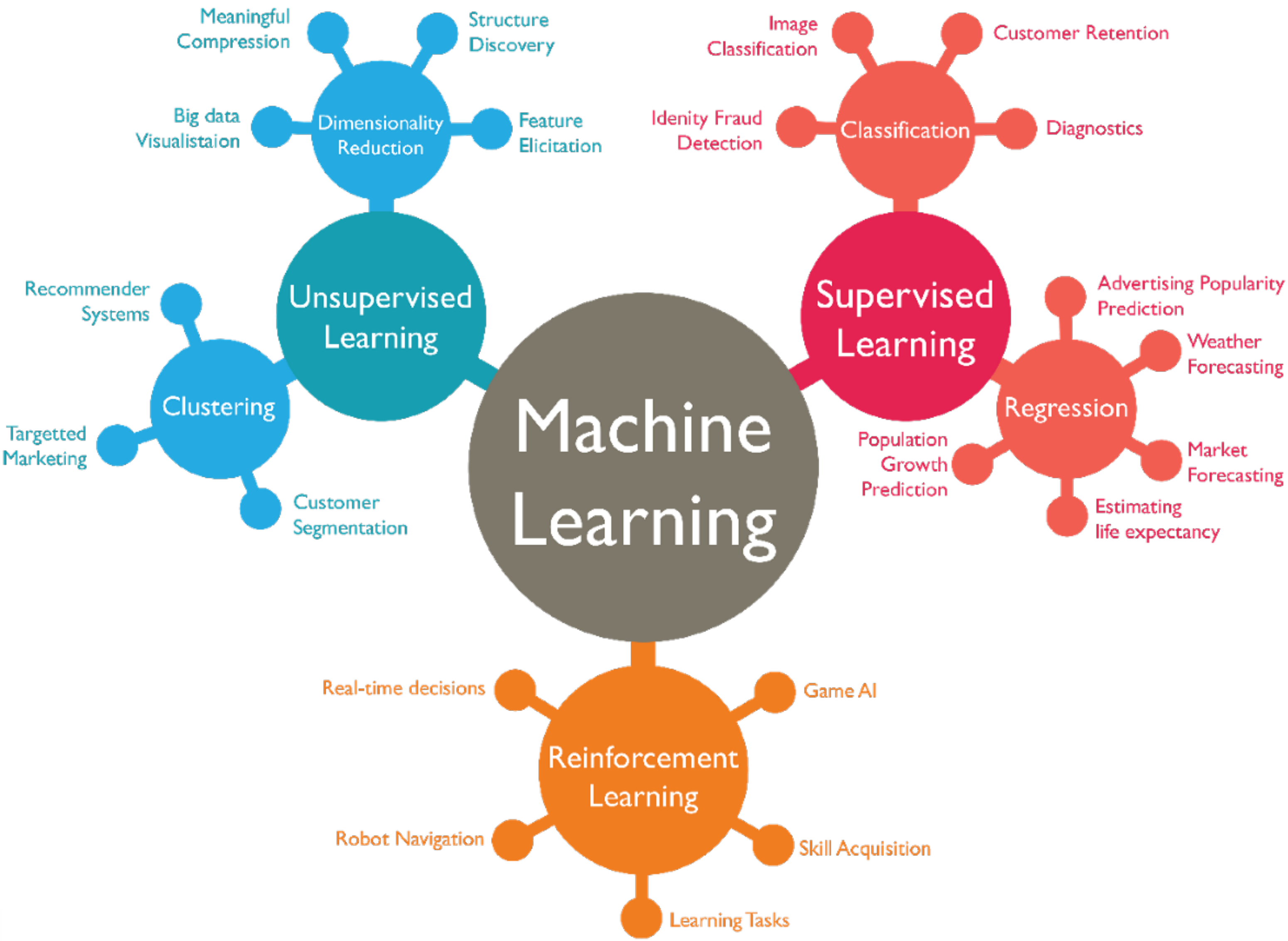
- ML is concerned with building computer programs that automatically improve with experience by:
  - **extracting knowledge from observations**
  - **using that knowledge to produce an "intelligent" response or behavior**
- ML is a field of Artificial Intelligence (AI) which "*refers to machines that perceive their environments and take actions to maximize their chances of achieving their goals.*" – Wikipedia

https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

Types of machine learning

- ML approaches vary in how they balance exploration of previously unknown knowledge and exploitation of current knowledge
- Classical approaches to ML include:
  - **Supervised learning** uses explicit guidance (supervision) of what to learn
  - **Unsupervised learning** represents the discovery of previously-unknown knowledge
  - **Semi-supervised learning** combines limited supervision with no supervision to enhance knowledge discovery
  - **Reinforcement learning** is based on maximizing cumulative reward

Types of machine learning



https://7wdata.be/visualization/types-of-machine-learning-algorithms-2/

Key components to a machine learning system

- Observations/experience: commonly presented as data (numbers, text, images, bank records, health records, etc.)
- Knowledge extraction: a computational technique to pull out knowledge from data
- Optimization method: a formula to optimize an objective function using the above knowledge
- Interaction protocol: an interface to present the "intelligent" behavior or response resulting from the above optimization

Supervised learning*

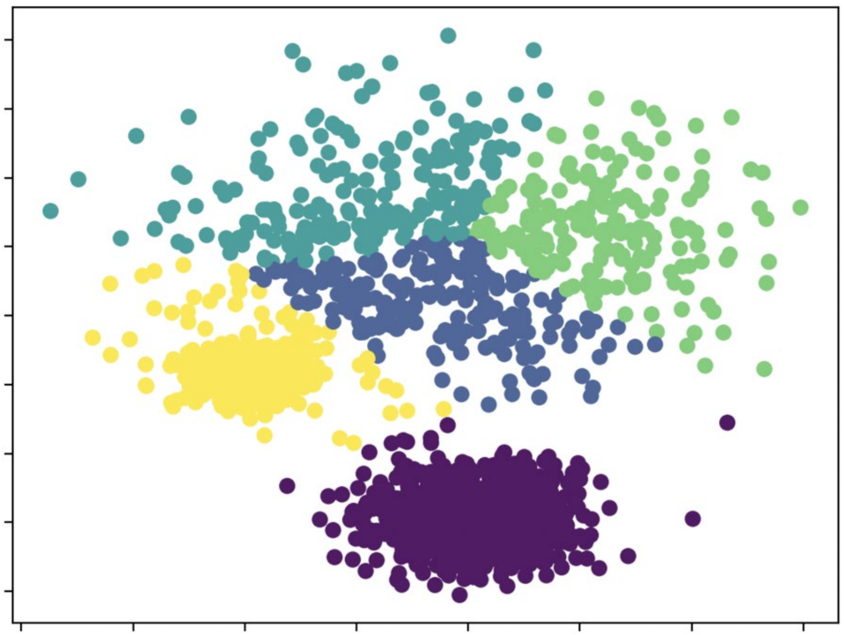
The training data consists of:

- data points (rows) list values (numeric or discrete) for each of the features (columns, e.g., height, weight, smoking status).
- rows are assigned outcomes:
  - If the outcome represents discrete categories (classes, e.g., positive or negative, benign or malignant), the task is called a *classification*
  - If the outcome is continuous (e.g., life expectancy or tolerable dose of medication), the task is a *regression* (which can also be used to produce discrete classes by using thresholds)

* Sidey-Gibbons, J., Sidey-Gibbons, C. Machine learning in medicine: a practical introduction. *BMC Med Res Methodol* **19**, 64 (2019). https://doi.org/10.1186/s12874-019-0681-4

Unsupervised learning

- There is no specified outcome, and thus, the task is to find a novel grouping of training data points that form clusters
- Data points have similar distances to the center of the cluster they belong to
- For a new data point, the task is to determine which cluster it belongs to (closest distance)
- The process is exploratory and highlights the discovery of novel clusters in the data



https://www.r-bloggers.com/2021/04/cluster-analysis-in-r/

Semi-supervised learning

- Labelling the training data with outcome can be tedious and expensive
- In this case, only a limited amount of training data is labelled but most of the training data is unlabeled (no outcome)
- Leveraging the labelled data, semi-supervised learning determines the outcomes for the unlabeled data, then constructs a learning model
- Example: in a large-scale systematic review on a specific topic, a search query yields a large number of documents. The task is to classify them as relevant or not to the systematic review. Human reviewers were able to label only 10% of the documents. A machine learning model can successfully label the remaining abstracts.

https://www.sciencedirect.com/science/article/pii/S0933365710001247#!

Reinforcement learning

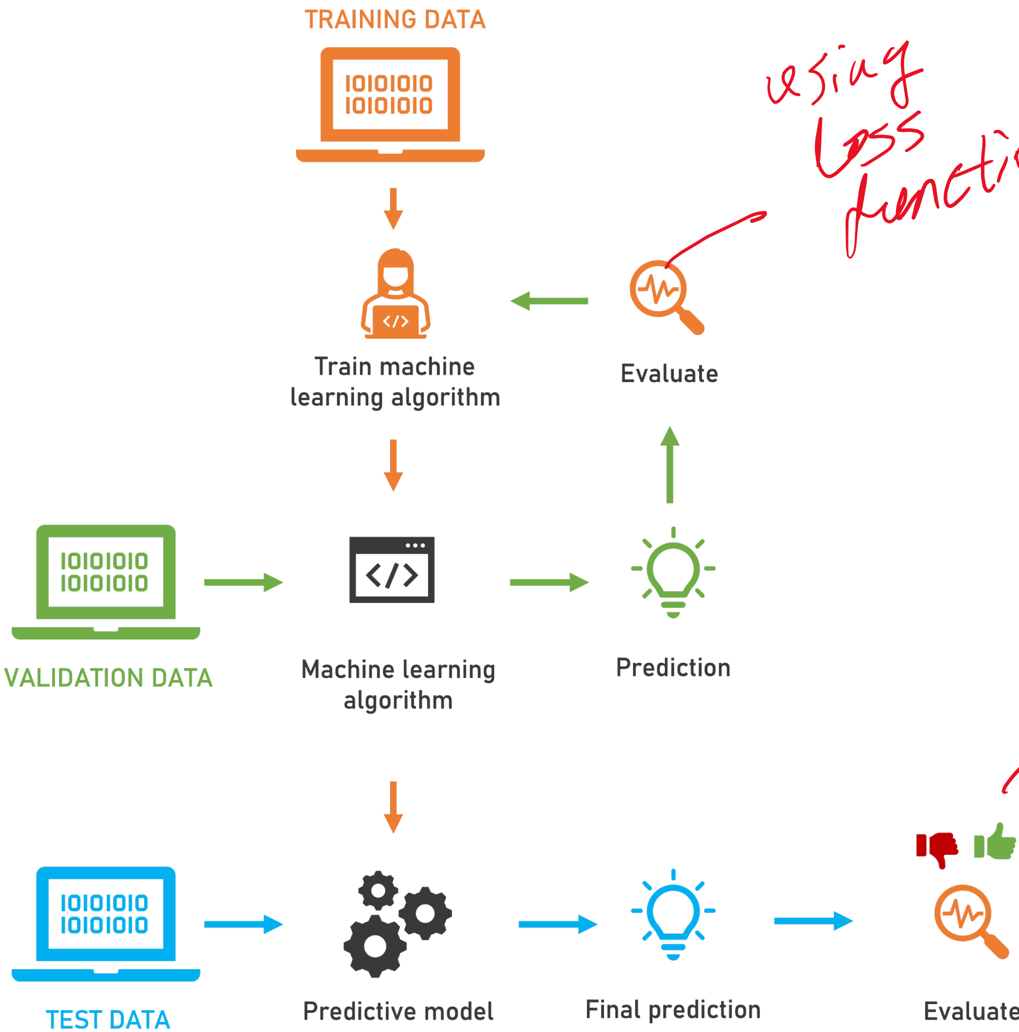
- The concern is to train a learning model to make a sequence of decisions to achieve a goal in a complex environment
- It is mostly geared towards the control of complex systems like self-driving cars
- The idea is to maximize the cumulative reward and minimize penalties based on a reward policy for a sequence of actions
- Example: learning to run for the development of prosthetic legs:

https://deepsense.ai/learning-to-run-an-example-of-reinforcement-learning/

Part II: Learning by minimizing loss

- Loss functions
- The trade off between bias and variance
- Regression Loss
- Classification Loss
- Gradient descent

Loss functions



using loss function

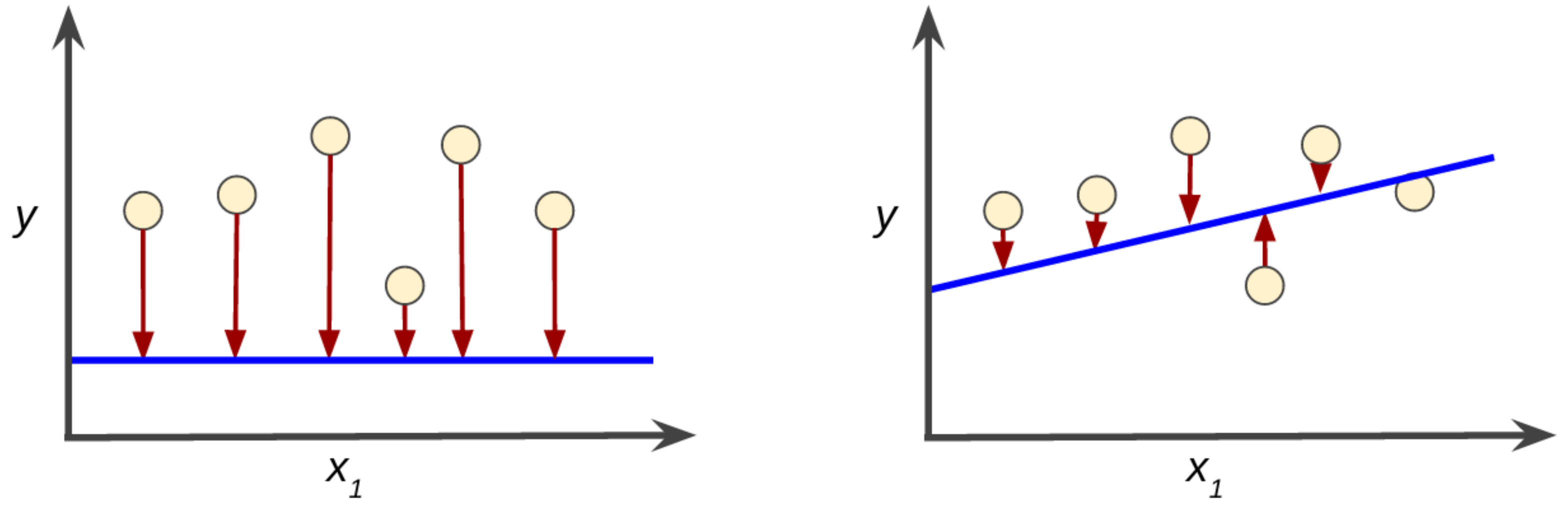
may be different loss function

Loss functions

- Loss functions help measure the gap between estimated values and the true values.
- A machine learning method can iteratively minimize a loss function to achieve better prediction performance
- Different learning tasks have different loss functions to achieve various goals
- The loss function used during training can be different than the one used during evaluation – the one used for evaluation should ideally be more interpretable / meaningful to the end use

https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23

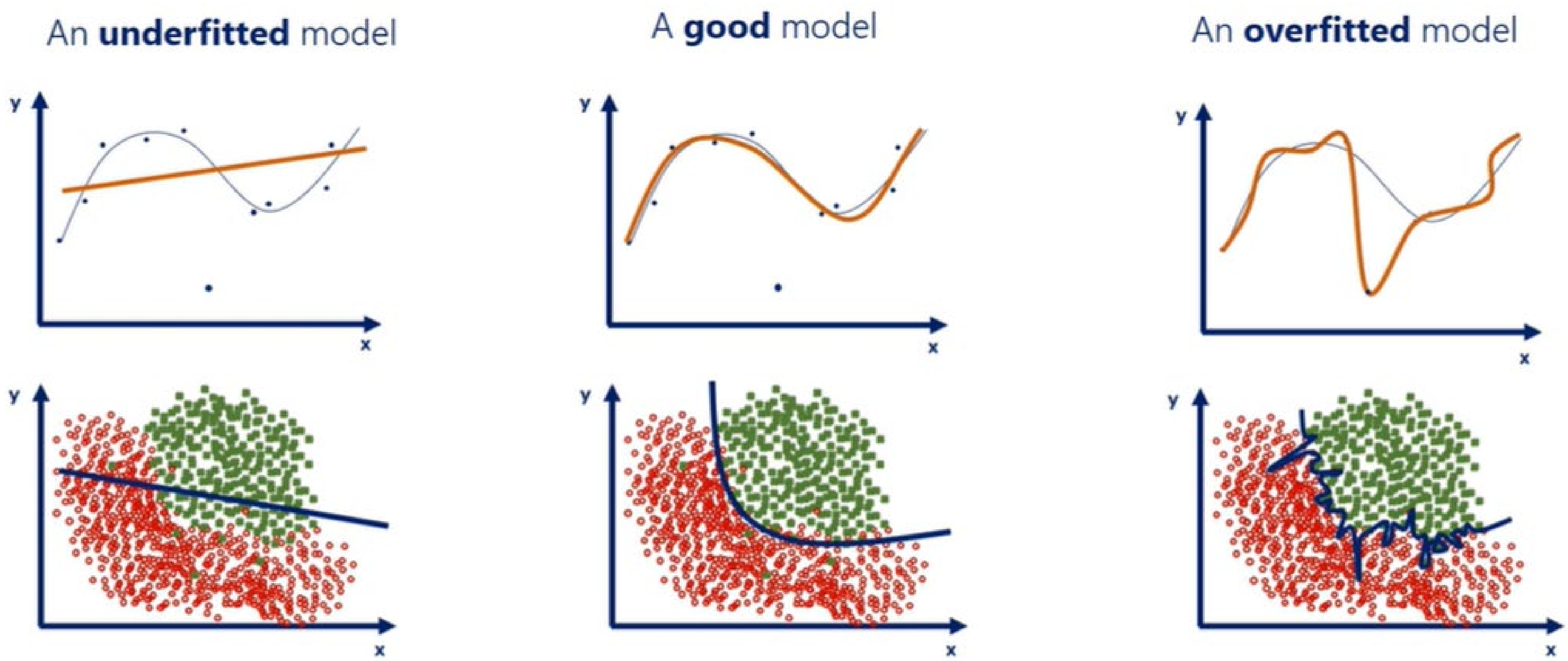
High loss (left) and low loss (right)



- The blue line represents predictions, the red arrows represent loss.

- Figure is from https://developers.google.com/machine-learning/crash-course/descending-into-ml/training-and-loss

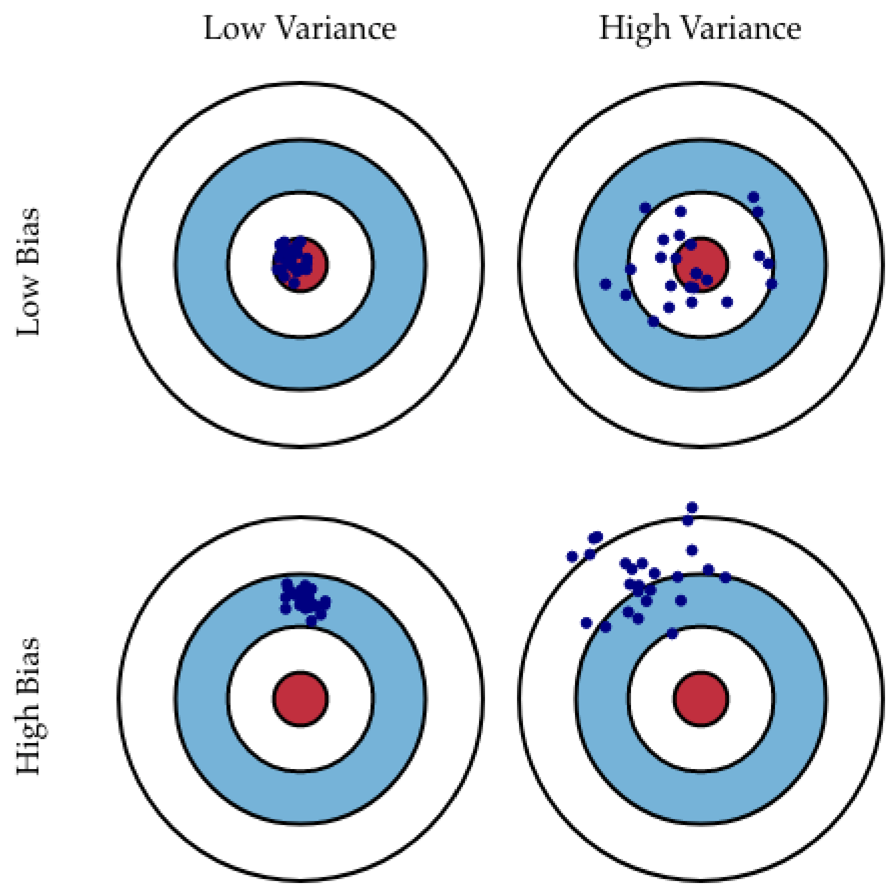
Overfitting and Underfitting



https://365datascience.com/tutorials/machine-learning-tutorials/overfitting-underfitting/

Bias & Variance

- Bias relates to how far the predicted values are from the correct values
- Variance is the degree to which the predictions vary between iterations of the model trained on different subsets of the data
- Errors due to bias: the model oversimplifies the task and ignores some of the training data (ignores features, learns the wrong target) – model underfitting
- Errors due to variance: the model “memorizes” the training data and fails to generalize to data it hasn’t seen before – model overfitting



- The center is the perfect model, which predicts the correct values
- As we move away from the center, predictions get worse

- Example and diagram are from: https://scott.fortmann-roe.com/docs/BiasVariance.html

Improve Bias

- Use a more complex type of model (e.g., a more complex artificial neural network)
- Use more features in the model
- Increase the size of the training dataset
- Reduce regularization (this is when an additional penalty is added to a model to avoid overfitting)

Improve Variance

- Fewer features (feature selection)
- Simplify the model
- Ensemble methods (combining machine learning models)
- More regularization

Regression loss

- Regression is predicting continuous values
- For observed value y_i, predicted value $\hat{y}_i$, and *n* observations in the data set, the Mean Absolute Error (MAE) is

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|$$

- MAE is also known as L1 loss and is used for non-Gaussian regression problems

Regression loss

- For Gaussians regression problems, Mean Squared Error (MSE) is the average of the squared differences between the actual and the predicted values.
- For observed value y_i, predicted value $\hat{y}_i$, and *n* observations in the dataset, the MSE is:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

- MSE is also known as L2 loss.

Regression loss

- Mean Bias Error (MBE) is used to calculate the average bias in the model
- MBE is the actual difference between target and predicted values (not absolute difference)
- Positive and negative errors could cancel each other out!
- For observed value y_i, predicted value $\hat{y}_i$, and *n* observations in the dataset, the MBE is:

$$MBE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)$$

Classification loss

- Classification is the task of predicting discrete class labels (outcome) for new data points.
- Binary classification has exactly two classes
- Entropy measures randomness in information, and cross entropy is a measure of difference in randomness between two random variables.
- Higher divergence of predicted probability from labels results in higher cross-entropy log loss:

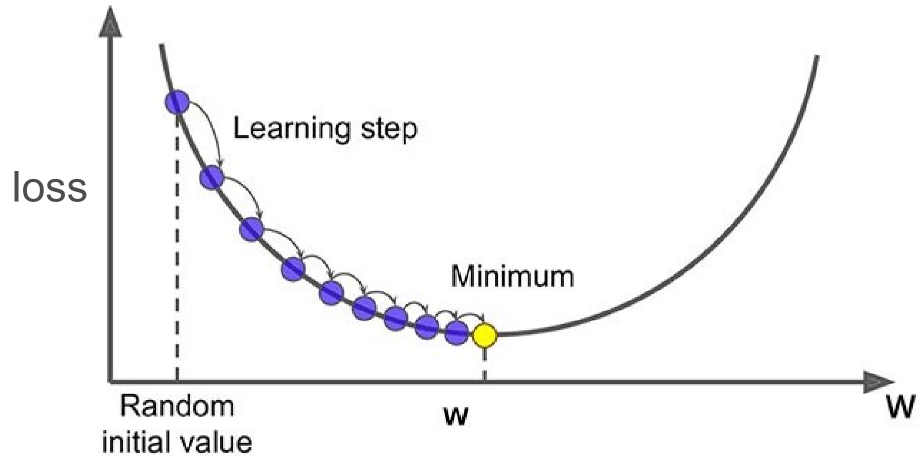
$$L_p = -\frac{1}{n}\sum_{i=1}^{n} y_i \times \log\left(p\left(y_i\right)\right) + \left(1 - y_i\right) \times \log\left(1 - p\left(y_i\right)\right)$$

- Where y_i is the true label and $p(y_i)$ is the predicted probability of being in the positive class.
- For binary classification (0 or 1), only y_i or $(1-y_i)$ can exist.

https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

Gradient Descent

jupyter
gradient_descent



- Regression yields convex loss vs weight plots
- Pick an initial value (a starting point) for w_1.
- The gradient of the loss is equal to the derivative (slope) of the curve and tells you which way is "warmer" or "colder."
- The gradient points in the direction of steepest increase in loss
- The gradient descent algorithm takes a step in the direction of the negative gradient in order to reduce loss as quickly as possible.

https://developers.google.com/machine-learning/crash-course/reducing-loss/video-lecture
https://www.r-bloggers.com/2017/02/implementing-the-gradient-descent-algorithm-in-r/