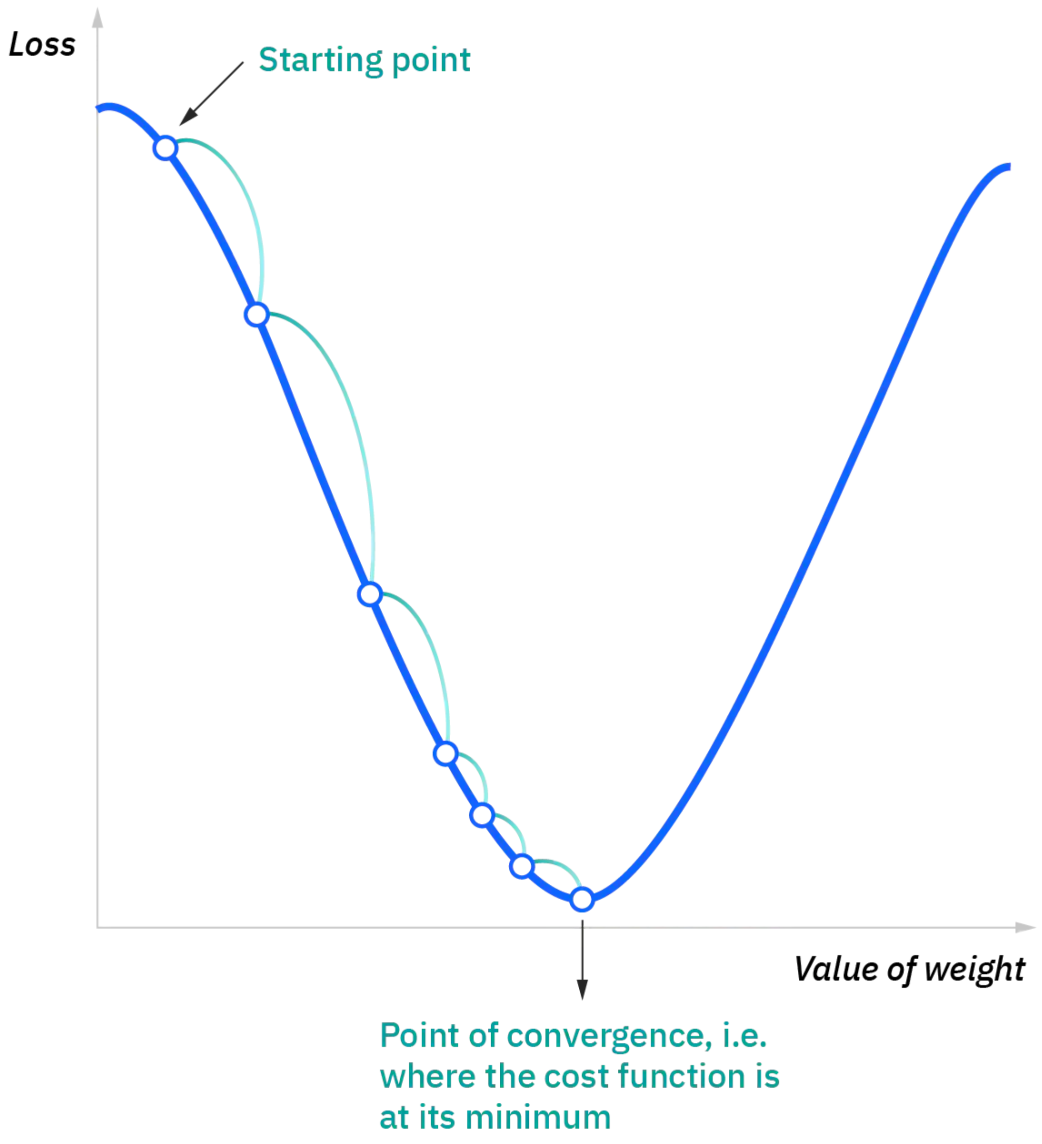
Module 2

Classification Problems

▼ What is gradient descent used for?

- to find the values of parameters of a function (f) that minimizes a cost function (cost)
- to find the steepest local minima in an n-dimensional dataset by updating the parameters of your model using the learning rate parameter to optimize a function initialized from that dataset
    - Parameters refer to coefficients in linear regression and weights in neural networks


Loss
Starting point
Value of weight
Point of convergence, i.e. where the cost function is at its minimum

▼ When should gradient descent be used?

it is best used when the parameters cannot be calculated analytically (e.g. using linear algebra) and must be searched for by an optimization algorithm

▼ What are cost functions used for?

- to estimate how badly models are performing
- it is a measure of how wrong the model is in terms of its ability to estimate the relationship between X and y

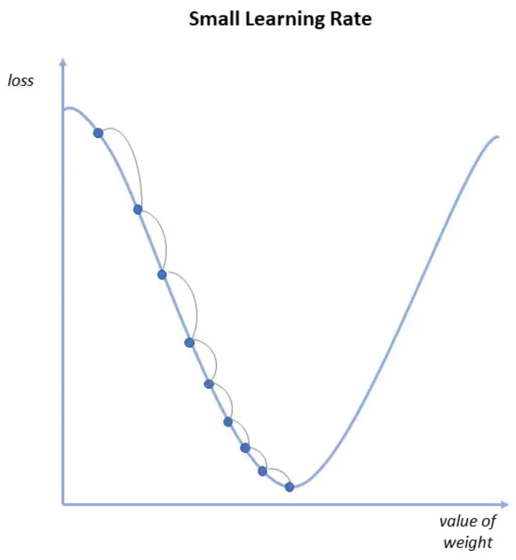
  - this is typically expressed as a difference or distance between the predicted value and the actual value

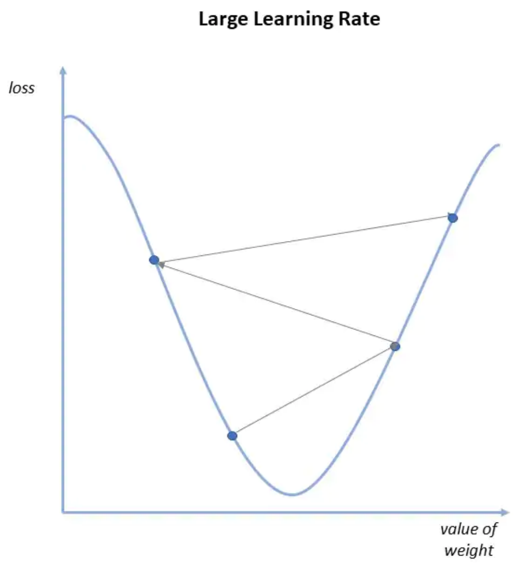
▼ What is the learning rate?

the size of these steps that are taken to reach the minimum

▼ Pros and cons of different learning rates

- With a high learning rate we can cover more ground each step, but we risk overshooting the lowest point since the slope of the hill is constantly changing.
- With a very low learning rate, we can confidently move in the direction of the negative gradient since we are recalculating it so frequently.
- A low learning rate is more precise, but calculating the gradient is time-consuming, so it will take us a very long time to get to the bottom.





▼ What are eager vs. lazy learning models?

- ***Eager*** learning models begin learning as soon as data is fed into it, rather than waiting for sample data as ***lazy*** models do.

- In practice, this means that ***lazy*** models train more quickly but are slower at actually classifying, making them difficult to use for real-time applications.
- ***Eager*** models, on the other hand, are well suited for real time prediction.

▼ What is K-Nearest Neighbor (K-NN) classification?

- a non-linear classification method that is able to create complex decision boundaries
- K-NN models are parametric, slow learners, and lazy
- K-NN does not make any assumptions about the shape of the decision boundary

▼ What is a parametric model?

all information necessary is contained within the parameters of the model

▼ What is Naive Bayesian classification?

- Classification models using Bayes theorem **use the assumption of independence between each feature pair**
- The independence assumption means, in practice, that all input variables are stochastically independent of each other
- It has advantages, such as being incrementally updatable; simple and easy to implement; and it can handle a large number of attributes and classes
- It is eager and fast, thus being well-suited to real-time prediction
- It is commonly used in spam filters, disease prediction, document classification, and sentiment analysis projects

▼ What do Naive Bayes classifiers and Support Vector Machines (SVMs) have in common?

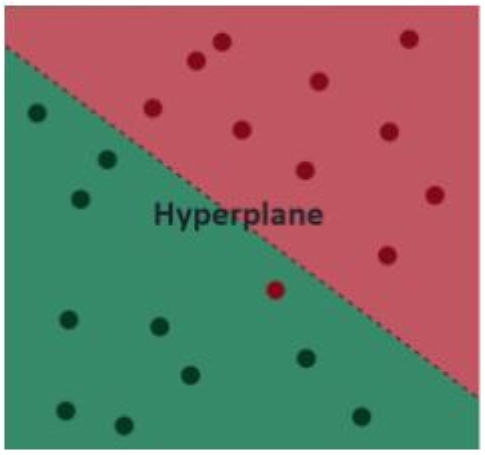
- They are both distance-based classifiers that use some form of the Fisher discriminant
  rule
  - finding Fisher's discriminant allows for an optimally accurate representation of data of n dimensions projected as a line in a lower dimensional space

▼ What is overfitting?

- overconfidence
- If your model is overtrained on a particular dataset, it may have focused on features that are too general or too specific for real world samples. Because of this, without first evaluating and testing your classifier on test sets, your classifier may do amazing on the training set and give completely inaccurate results when presented with real-world problems.

▼ When is classification linear?

- when the decision boundary is a linear function, meaning that a hyperplane separates the decision boundary between the feature space of the two (or more) classes
- you can often visually identify a linear solution by the fact that a straight line can be used to divide the decision boundary (through the origin) in a two-dimensional graph of the data.



▼ What should you evaluate your model with?

a subset of the original training data

▼ What categories should a dataset be divided into?

- Training, validation, and testing
- Common percentage ratios: 70/15/15, 80/10/10, and 60/20/20

▼ What does validation do?

- It prevents overfitting by tuning parameters during the training process by comparing the models outputs from the training data to outputs from the validation set.
- The labeled test set is then used after training to provide an accuracy estimate as a percentage of test samples that were correctly guessed.

▼ How do K-NNs work?

- The input is compared to its "neighbors" to find ***k*** number of data points most similar to it.
- ***K*** is defined beforehand.
- The class distribution of ***k*** is then used to "vote" on the most likely class the new input belongs to.

▼ What is the general process for building a K-NN classifier in Python?

1. Load your dataset;
2. Initialize the value of ***k***, which is the number of neighbors that the model will consider (this number must be odd to prevent ambiguity);
3. Calculate the distance between the input and every other data point, and select ***k*** data points that have the shortest distance from the input value;
4. Output returned is the mode of the combined labels of ***k***.

▼ What is Principal Component Analysis (PCA) used for?

to reduce the dimensionality of data

▼ What are decision trees?

- A decision tree is a classification tool related to intelligent search.
- How: resembling a flow-chart, the input is run through a series of nodes, each of which passes the input to the next node based on predefined conditions or weights in a process called recursive binary splitting. Each node or leaf represents a possible class label, and the junctions between nodes represent features.

- Decision trees are unfit for incremental learning or generalized datasets or datasets with complicated or arbitrary decision boundaries
- They are computationally efficient and support linear and non-linear functionality.
- They can be eager or lazy.

▼ What are Support Vector Machines (SVMs)?

- Using a math trick known in the field as "the kernel trick", nonlinear SVMs such as Radial Basis Function (RBF) create new features by squaring each existing feature to expand the data into a higher-dimensional space, allowing the hyperplane to be drawn, before collapsing the data back into a lower-dimensional space.
- SVMs work well with small training datasets and large numbers of features and can be highly accurate.
- SVMs are examples of eager models.