1 Introduction

Artificial Intelligence

- broad concept
- different interpretations
- we do not have a definition of inteligence

Statistical machine learning

- Algorithms and applications where computer learn from data

AGI

- Artificial General Intelligence
- Hypothetical computer program that can perform intellectual tasks as well as, or better than a human.

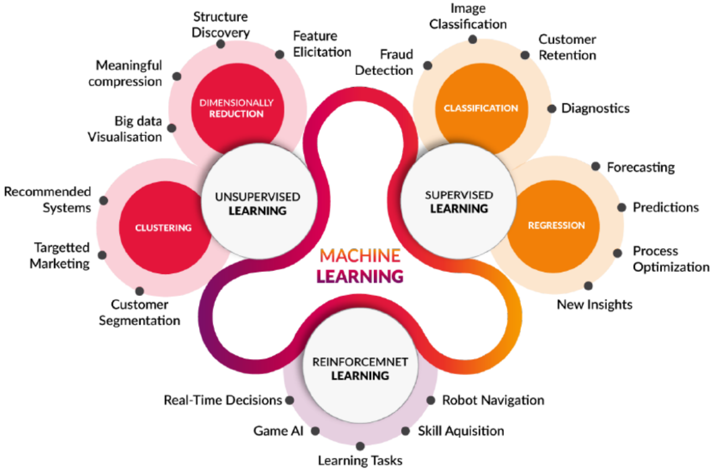
Turing Test

- Also called imitation game
- Tests of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from that of a human
- Has some philosophical problems (Complex problems, humans cant solve / AI must learn to lie)

Examples of application (today):

- Personalization of news feeds
- Product searching and recommendation s on eCommerce platforms
- Voice-to-text
- Predictive maintenance

1.1 Tasks and Algorithms of Machine Learning



1.2 Natural Language Processing (NLP)

- Automated processing of human language (written & spoken)
- Aims to understand and generate human (natural) language
- Understanding spoken text is still difficult
- Understanding written text became BIG business (search-engines)
- Generating human-like conversations is still very hard

1.3 Dialogflow

Intents

- Recognizes the need of a user
- Require training to match to user inputs
- Follow up Intents (on Success)
- Fallback Intents (on Failure)

Entities

- Extract information from user inputs
- Help to identify required intent
- System Entities: (Date and time / Numbers / Amounts / Units / etc.)
- Developer Entities: defined by list of words (@pizza-type / @drink / etc.)
- User Entities: transient, temporary Information based on Conversation

Dialog

- Linear: Gather a list of information
- Non Linear: Using Contexts

Context

- Each Intent can have Input & Output Context
- Intents are active based on active Context
- Expire automatically

Fulfillment

- Action triggered on fullfilled Intents
- e.g. Webhook

1.4 7 Steps of ML

1. Gathering data
2. Preparing that data
3. Choosing a model
4. Training
5. Evaluation
6. Hyperparameter tuning
7. Prediction

2 Natural Language Processing (NLP)

2.1 Ingredients of Machine Learning

1. Data

- Dataset
- Pre-Processing Pipe-Line including cleansing, feature-engineering, data augmentation etc.

2. Cost-Function (Loss)

- Formal mathematical expression for good / bad
- Commonly Mean Squared Error (MSE)

3. Model

- From linear model: $\hat{y_i} = ax_i + b$
- To complicated million parameter neural networks
- Different tasks require different models (regression / decision tree)

4. Optimization Procedure

- Algorithm that changes the parameters of the model that the cost-function is minimized.
- E.g. Stochastic Gradient Descent (SGD), ADAM, RMSProp...

2.2 More ingredients

For successful ML, there are many more ingredients:

5. Performance optimization

- Building of efficient pipe-lines
- Folowwing tool specific recommendations

6. Visualization and evaluation of the learning Process

- Learning curves
- Performance measures
- Tensorboard

7. Cross-Validation & Regularization

- Train models that generalize well to unseen data
- Estimate the generalization error

2.3 Representation of Words

Vectors can be used to represent words based on their meaning.

2.3.1 One-hot representation

- Vector with a single 1-Value
- All other Values are set to 0
- Count the Number of different Words, Define one unique vector per word:

Dini Mom isch fett.

$$\text{Dini: } \begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix} \quad \text{Mom: } \begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix} \quad \text{isch: } \begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix} \quad \text{fett: } \begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix} \quad \text{'.': } \begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix}$$

Disadvantages:

- Very high dimensional vector space (1 Dimension / unique Word)
- Sparse Representation: Eech vector has a single 1 and N Zeroes. (Memory Inefficient)
- No Generalization: All words are unrelated to each other.
- Does not capture any aspect of the meaning of a word

2.3.2 Indexing

Make a list of words (optionally alphabetically). Use the index to represent each word.

Example:

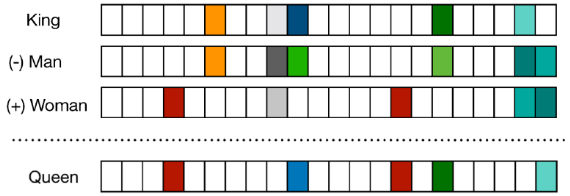
Dini Mom isch fett.

Dini: 0, Mom: 1, isch: 2, fett: 3, '.': 4

- Dense Equivalent of one-hot encoding
- Indexes are not more useful that one-hot vectors
- Often used as preprocessing step
- Indices / One-Hot Vectors are fed into a network which learns more useful representations

2.3.3 Distributed Representation

- Words that occur in similar contexts (neighboring words) tend to have similar meanings
- Similar words share similar representations
- Distributed representations can be learned



Words to Vectors:

- Mathematical function maps word to high dimensional Vector
- In neural networks, this function is implemented in the Embedding Layer

Advantage of Vectors

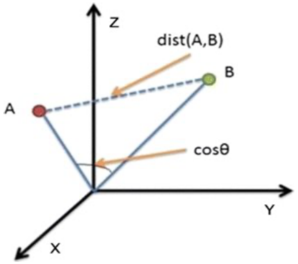
- Good embedding maps simiar/related words to similar regions of the vector space
- Dot-Product (Skalarprodukt) is a measure of similarity
- Possible to add/subtract vectors

Calculate Similarities between words Dot-Product (Skalarprodukt) of 2 Vectors:

- maximal when parallel (0°) (1 with norm (length) 1)
- zero when orthogonal (90°)
- minimal (negative) when opposite directions (180°) (-1 with norm (length) 1)

Cosine Distance

- Way to calculate how similar two words (vectors) are



$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

3 Probability

3.1 Random Variables

- Values depend on outcomes of a random phenomenon
- Random variable X is a variable that takes a numerical value x, which depends on a random experiment
- **Discrete:** X takes any of a finite set of values $1.5, 2.123, 6.2, 10$
- **Continous:** X takes any alue of an uncountable range e.g. real numbers from an interval

Best we can know

- All possible values
- Probability of each value

E.g. The discrete random variable X is the number observed when rolling a fair dice.

$Pr(X = x)$ / $P(x)$: 1/6 for each possible value

3.1.1 Two random variables

Joint Probability

- Joint Properties of two random variables
- Defined by the Joint Probability Mass Function

E.g. Dice1 $=5$ AND Dice2 $=4$

$P_{XY}(5, 4) = 1/36$

	X=1	X=2	X=3	X=4	X=5	X=6
Y=1	1/36	1/36	1/36	1/36	1/36	1/36
Y=2	1/36	1/36	1/36	1/36	1/36	1/36
Y=3	1/36	1/36	1/36	1/36	1/36	1/36
Y=4	1/36	1/36	1/36	1/36	1/36	1/36
Y=5	1/36	1/36	1/36	1/36	1/36	1/36
Y=6	1/36	1/36	1/36	1/36	1/36	1/36

Independant random Variables

- Joint Probability is the product of the individual probabilities

$P(X,Y) = P(X) * P(Y)$ (only if independant)
$P(X,Y,Z) = P(X) * P(Y) * P(Z)$ (only if independant)

Correlated random Variables

- There are events that are not independant
- Such random variables are correlated
- X: observe clouds (0=no, 1=small, 2=big)
- Y: observe rain (0=no, 1=light, 2=moderate, 3=heavy)

Conditional Probability

- One variable is no longer random
- X is observed, its value is fixed
- Calculate the probabilities of Y given X: $P(Y|X)$

$P(X,Y) = P(X|Y) * P(Y)$
$P(X,Y) = P(Y|X) * P(X)$
$P(Y|X) = \frac{P(X,Y)}{P(X)}$

Bayes Rule

$P(X|Y) * P(Y) = P(Y|X) * P(X)$
Therefore:
$P(Y|X) = \frac{P(X|Y)*P(Y)}{P(X)}$

4 Python

Chani alles

5 Data Visualization

- See trends, clusters and patterns in data
- Difficult to see in raw data
- Detect outliers and unusual groups
- Validate Hypothesis/Conjecture/Theory

Important in a Plot:

- X-Axis / Y-Axis
- Title
- Scale
- Dimensionality of the data 2D / 3D

5.1 Data Analysis Libraries

5.1.1 NumPy

- Package for scientific computing in Python
- Multidimensional array object
- Routines for fast array operations (sorting, selecting, FFT, linalg, etc)

5.1.2 pandas

- Built on top of NumPy
- Routines for accessing tabular data from files (.csv, xls, etc.)
- Supports 2-dimensional data (dataframe and series)
- Dataframes are something like database tables

5.1.3 MatPlotLib

- Library for visualizing data
- Bargraphs, Histograms, Piecharts, Scatter plots, lines, boxplots, heatmaps, etc.

5.1.4 Seaborn

- Extension of MatPlotLib, NumPy and pandas
- More user friendly
- Plots are aesthetically better

5.1.5 Chart types

Line Plots

- Bivariate, Continous
- Recognizes trend (pattern of change)

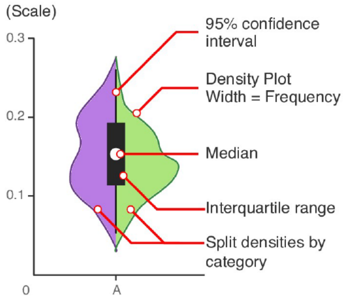
Bar Chart

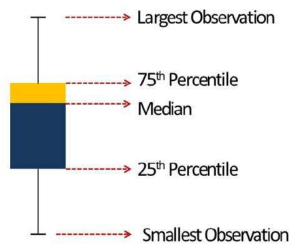
- Used for categorical data
- Counting based on each category

Histogram

- Represents the empirical distribution of a variable
- Automatically creates bins (interval) along the range of values
- Shows vertical bars to indicate the number of observations / bin

Descriptive Statisics: Box Plots and Violin Plots





Scatter Plot

- Relationship between continous variables
- Helps to get an idea of the degree of correlation between variables

6 Regression

6.1 What is a model?

In ML, we use the term **model** for any mathematical function that explains the data:
$y_i = f(x_i)$
$y_i = f(x_i) + \epsilon_i$
where ϵ_i is unexplained noise. It is often assumed that ϵ_i follows a normal distribution.
Instead of approximating y_i, we calculate an **estimate** $\hat{y_i}$ (y hat) of the usually unknown y_i:

$$\hat{y_i} = f(x)$$

6.1.1 Linear Regression

- Only consideres a linear relationship between input and output
- In the simplest case, x and y are scalars and the linear model therefore has only two free parameters
- The goal is to identify a (slope) and b (intercept) for which the linear model best explains the data

$$\hat{y_i} = ax_i + b$$

6.1.2 Mean Squared Error (MSE)

- Loss we want to minimize
- Usually divided by 2

$$\hat{y_i} = ax_i + b$$
$$e_i = y_i - \hat{y_i}$$
The difference e_i, called residual
$$E = \frac{1}{2N} * \sum_{i=1}^{N} e_i^2$$
$$E = \frac{1}{2N} * \sum_{i=1}^{N} (\hat{y_i} - (a * x_i + b))^2$$

6.1.3 Correlation and Causality

- Correlation is not causality
- Correlation refers to the degree to which a pair of variables are linearly related
- Linear regression is a tool to detect correlations between two or more variables
- Correlation can be quantified using the Pearson correlation coefficient

7 Optimization

- Training or learning in AI often suggests an algorithm performing some sort of optimization
- It is the problem of finding a set of inputs to an objective function that results in a maximum or minimum function evaluation
- In our examples the objective is to minimize the loss function

7.1 Gradient Descent

- Iterative Method
- Each iteration, the model parameters are updated such as that the Loss (MSE) is reduced

7.2 Stochastic Gradient Descent (SGD)

- At each iteration, the gradient is calculated on a (randomly selected) subset of the data
- For a fixed learning rate, SGD does not converge

7.2.1 Annealed SGD

- The learning rate alpha is reduced over time
- This is called (simulated) annealing
- There are different options (called schedules) how to reduce alpha over time

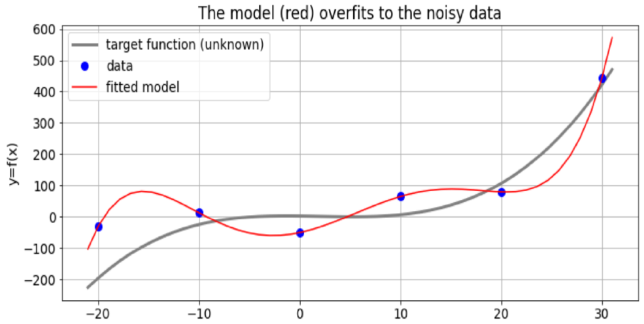
7.2.2 General remarks on SGD

- Gradient-based methods only work if we can express a Loss function as a differentiable function
- SGD is dealing woth only a single datum at each iteration. This is very inefficient and rarely used.
- Batch- or mini-batch gradient-descent is usually used

8 Generalization & Regularization

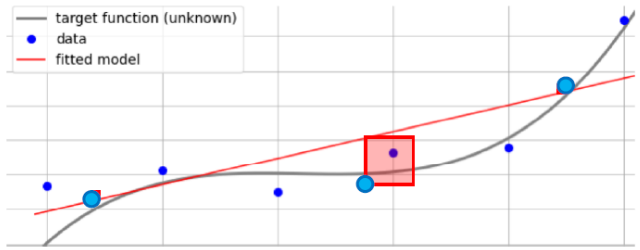
8.1 Overfitting

- A model that perfectly fits the data does not have to be perfect
- In-Sample Error (Trainig error) was minimized (MSE = 0)
- Out-of-sample Error (Generalization Error, Test Error) is the MSE of new Data
- A good model has a low Generalization Error
- Overfitting happens if the MSE of Training Error is small thanks to a complex model but the Generalization Error is large



8.2 Underfitting

- Using a too simple model
- In-Sample Error is large
- Generalization Error is large



8.3 Training-Set, Test-Set, Model Evaluation

- The Generalization Error can't be calculated
- But Estimated
- Split the data into 2 sets
  - Training-Set (80% of data)
  - Test-Set (20% of data)

Training:

- Fit the model to the training set
- This minimizes the in-sample error

Evaluating

- Using the Test-Set
- Produces the Test-Error
- This is an estimate of the Generalization Error

8.4 Bias-Variance Trade-off

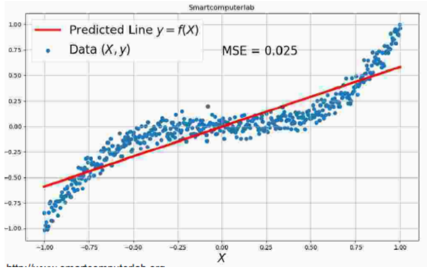
Variance: Difference of fits between data sets.
Bias: Results that are systematically prejudiced due to faulty assumptions.

High Bias

- A too simple model for the given data

Low Variance

- The model is relatively stable
- Very simular model if trained with new data

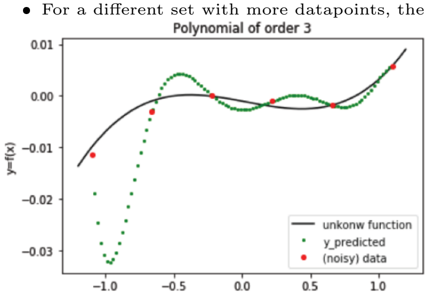


Low Bias

- A more complex model can better explain the data

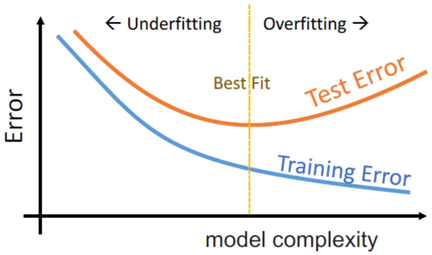
High Variance

- Given a new datapoint, the MSE can be very large
- For a different set with more datapoints, the model may be very different



8.4.1 Trade-off

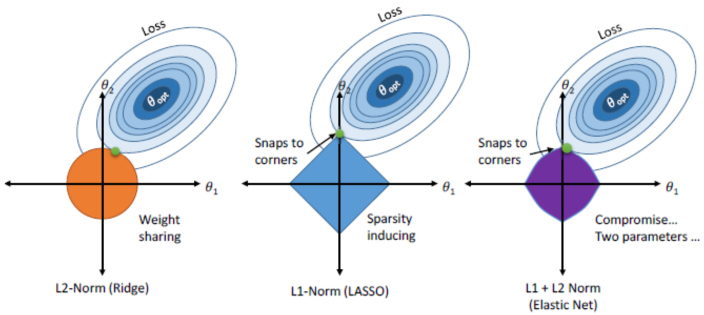
- Higher bias implies lower variance
- Lower bias implies higher variance
- In practice, all we want is low variance
- The model can only be as complex as the data permits
- You have to find an optimal balance between bias and variance



8.5 Regularization

- Technique to control the model complexity
  - Add a penalty term to the Loss
  - More complex models get a higher penalty
  - Add a constrain to the optimization process
  - *regularized loss = MSE + λ model-complexity*

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$



9 Cross-Validation

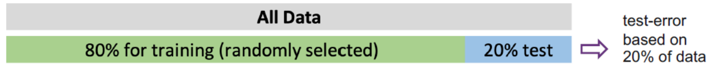
Problem with 80/20 Data Separation

- Test Error depends on random set
- For different Set, the test error would be different

With Cross-Validation we can obtain a better estimate of the generalization error

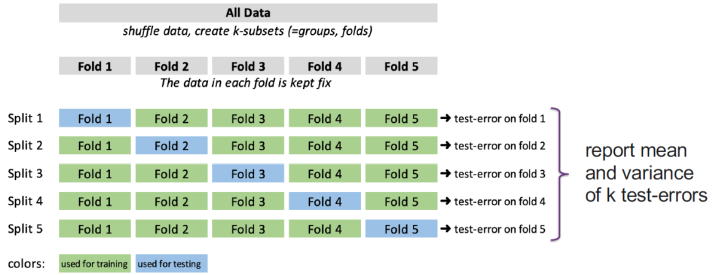
9.1 k-fold Cross-Validation

- Without cross-validation:



With k-Fold Cross-Validation

- The data is split once into k folds. Then train/test is repeated k-times. Each fold participates in k-1 training phases and is used once for testing:



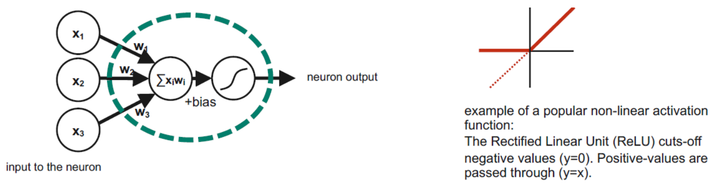
9.1.1 Some Comments

- Typical Values for k are 5,10 or N
- The data of a fold does not change during procedure
- Do not preprocess the whole dataset
- Apply the preprocessing pipe-line to each split

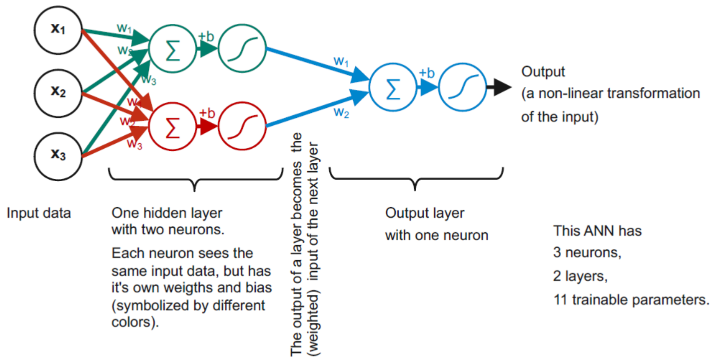
10 Artificial Neural Networks (ANN)

10.1 Artificial Neurons

- Receives an input vector $[x_1, x_2, \ldots]$
- Each neuron has its own input weights $[w_1, w_2, \ldots]$ and **bias** b
- Calculates the sum of the weighted input (dot product $\vec{x} * \vec{w}$), adds a bias b, and passes it through a nonlinear activiation function



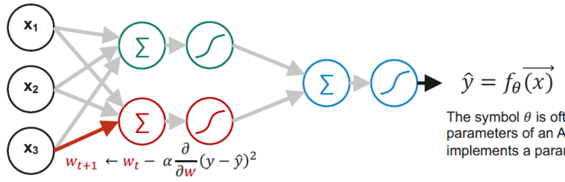
10.2 Simple ANN



10.3 Traning an ANN

Supervised learning

- For each input $\vec{x}$ we are given the output $\vec{y}$
- ANN is initialized with random weights
- An optimizer reduces a cost-function (e.g. MSE)
- At every iteration, and for every single weight w and bias b, the partial derivative needs to be calculated. (Backpropagation)



The symbol θ is often used to denote *all* parameters of an ANN. The ANN implements a parametrized function f_θ

11 Classification & Logistic Regression

11.1 Binary Classification

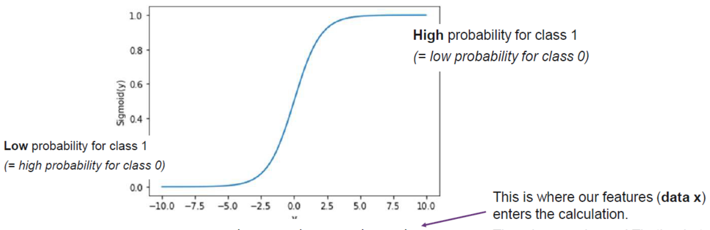
- Decision with 2 possible outcomes
- Hail in Lausanne (yes/no)
- Master admission (admission / no admission)
- Based on different data / entity

11.1.1 Decision using Linear Regression

- Train the model with gradient descent
- **Bad Idea!**
- Models the response (y) and post process the response to compute the probability

11.1.2 The sigmoid function

$$sigmoid(y) = \frac{1}{1+e^{-y}}$$



$$y = w_1x_1 + w_2x_2 + w_3x_3 + w_4 + \cdots$$

Probabilities
- We can write the estimated probability
- For a prediction we can write

$$P(x) = \frac{1}{1+e^{-(W^Tx)}}$$

11.1.3 Maximum Likelihood

- Given all the data points (X,Y) we want to maximize the probability that all the predictions are correct.
- For each of the training data, we want to maximize the likelihood of correct prediction
- We can use Gradient Descent to find W

12 Classifier Evaluation

12.1 Confusion Matrix

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP),	False negative (FN),
	Negative (N)	False positive (FP),	True negative (TN),

Source: Wikipedia

- True Positive (t_p):
  - model predicted "yes/positive", and the truth is also "yes/positive."
- True Negatives (t_n):
  - model predicted "no/negative", and the truth is also "no/negative."

Prediction Correct

- False Positives (f_p):
  - model predicted "yes/positive", and the truth is "no/negative".
- False Negatives (f_n):
  - model predicted "no/negative", and the truth is "yes/positive".

Prediction Wrong

Mean Accuracy:
- How often is the classifier correct?
- $A = (t_p + t_n)/n$

Mean Error:
- How often is the classifier wrong?
- $E = (f_p + f_n)/n$

Precision:
- When the prediction is 1, how often is it correct?
- $P = t_p/(t_p + f_p)$

Sensitivity, Recall, True Positive Rate (TPR):
- How often the prediction is 1 when it's actually 1
- $R = t_p/(t_p + f_n)$

Miss Rate, False Negative Rate (FNR)
- $MR = 1 - TPR$

12.2 Why Accuracy is not enough?

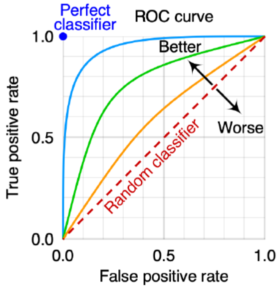
- If the prediction is constant the accuracy may still look decent
- E.g. allways predict false
- 90% of the data is false
- Accuracy = 90% (decent)
- Precision = 0
- Recall = 0

12.3 Precision vs. Recall

- Increasing precision reduces Recall and vice versa
- Threshold is a business decision (depending on goals)

12.4 Receiver Operating Characteristics

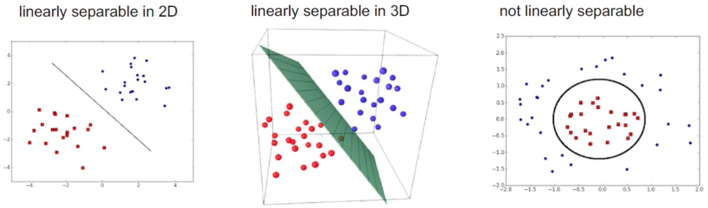
- Defined by FPR and TPR as x and y axes
- Visualizes tradeoff between TP (benefits) and FP (cost)



Area under the curve
- Area under the ROC curve
- Shows how well the TPR and FPR is looking in the aggregate
- The greater the area under the curve, the higher the quality of the model
- The greater the area, the higher the ratio of TP to FP

13 KNN

13.1 Linear Seperability



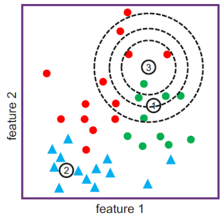
- Based on logistic regression model, you can draw a line
- This is the Linear decision boundary
- If a simple line perfectly seperates the classes, then the classes are said to be linearly seperable

13.2 Non-Linear decision boundary

- When classes are not linearly seperable
- Resort to polynomial terms

13.3 k-Neares Neighbors (KNN)

- A datapoint is know by the company it keeps
- Computes k nearest neighbours
- Returns the most frequent class of the k neighbours



	k=3	k=5	k=10
sample 1	g	g	g
sample 2	b	b	b
sample 3			

- Parameter: how many neigbours? Choice of k!

13.3.1 Distance Metric

- Cosine Distance
- Manhattan Distance
- Euclidean Distance (most used)
- Minkowski Distance

13.3.2 Advantages

- Easy and simple ML model
- Few hyperparameters to tune

13.3.3 Disadvantages

- k should be wisely selected
- Large computation cost during runtime if sample size is large
- Not efficient for high dimensional datasets
- Proper scaling should be provided for fair treatment among features

13.3.4 Hyperparameters

- **K Value**: how many neighbours to participate in the KNN algo.
- **Distance Function**: Euclidean distance is most used

14 Clustering

14.1 Unsupervised Learning

- We are given Data (features, x) wihout labels (y)
- Can we still learn something from the data?
- Yes! Often the data has some structure
- **The goal** of unsupervised learning is to self-discover patterns from the data

14.2 Clusters

- Data points which have shared properties
- Fall into one cluster or one alike group
- Similar Data Points are close together

14.2.1 Applications

- Social Network Analysis
- Astronomical Data
- Marked segmentation
- Recommendation systems

14.3 Naive K-means

1. Let us assume we know the number of clusters k_c
2. Initialize the value of k cluster centres (aka, means, centroids) $(\mathbf{C_1}, \mathbf{C_2}, \ldots \mathbf{C}_{k_c})$
3. Assignment :
   1. Find the **squared Euclidean distance** between the centres and **all the data points.**
   2. Assign each data point to the cluster of the **nearest centre.**
4. Update: Each cluster now potentially has a new centre (mean). Update the centre for each cluster
   1. New Centres $((\mathbf{C'_1}, \mathbf{C'_2}, \ldots \mathbf{C'}_{k_c})$ = Average of all the data points in the cluster$(1,2,\ldots,k_c)$
5. If some stopping criterion met, Done
6. Else, go to Assignment step 3

14.3.1 Stopping Criterion

- When centres don't change (time consuming)
- The datapoints assigned to specific cluster remains the same (takes too much time)
- The distance of datapoints from their centres >= treshold we have set
- Fixed number of iterations have reached (choose wisely)

14.3.2 Initialization

- Performance depends on the random initialization
- Some seeds can result in a poor convergence rate
- Some seeds can converge to suboptimal clustering
- If centres are very close, it takes a lot of iterations to converge
- Initialize randomly, run multiple times

14.3.3 Standardization of data

- Features with large values may dominate the distance value
- Features over small values will have no impact
- Normalize values!

14.3.4 Sklean k-means

Initialization
- Init = K-means++
- Only initialization of the centroids will change
- Chosen centroids should be far from each other

max_iter:
- Number of iterations before stopping

n_init:
- Number of time the k-means algorithm will be run with different centroid seeds

14.3.5 Evaluating Cluster Quality

- Make clusters so that for each cluster the distance of each cluster member from its center is minimizes

Inertia or within-cluster sum-of-squares (WCSS)

- Sum of squared distances to center
- As small as possible

Silhouette Score

- How far the datapoints in one cluster are from the datapoints in another cluster
- SS of a point: $\frac{b-a}{max(a,b)}$
- a: average intra-cluster distance (distance between each point within)
- b: average inter-cluster distance (distance between a cluster and its nearest neighbour)

15 Ensamble Methods

15.1 Wisdom of Crowd

- Suppose you have a difficult question
- Ask many people and aggregate the answer
- This might work very well instead of finding the best suited person

15.2 Ensamble

- Wisdom of Crowd can be applied to ML
- Instead of finding the best model, aggregate the results of weak models
- Aggregate predictions of regressors or classifiers
- Might get better accuracy than the best predictor
- Ensamble: group of predictors

15.3 Ensamble Method

- Suppose we have many different weak models (better than random)
- Get prediction from all of them and take a vote
- Class with most votes is the predicted class
- Commonly used towards the end of a project
- **Requirement**: enough models / diverse models

15.4 Bagging and Pasting

Bagging (Bootstrap Aggregating)

- Sampling with replacement
- Allows data points to be used several times

Pasting

- Sampling without replacement

15.5 No free lunch theorem

No single machine learning algorithm is universally the best-performing algorithm for all problems

15.5.1 Out of Bag (oob) Evaluation

- Using Bagging
- Some Data Points may not be used at all
- Use them for evaluation